DOCUMENT RESUME

ED 077 182 EM 011 010

AUTHOR	Hoggatt, Austin Curwood, Ed.
TITLE	1973 Winter Simulation Conference. Sponsored by ACM/AIIE/SHARE/SCi/TIMS.
PUB DATE	Jan 73
NOTE	916p.; Proceedings of the Annual Winter Simulation Conference (6th, San Francisco, California, January 17-19, 1973)
EDRS PRICE	MF-$0.65 HC-$32.90
DESCRIPTORS	Computer Programs; *Computers; *Conference Reports; Games; Game Theory; Management; Organization; Policy Formation; Simulated Environment; *Simulation

ABSTRACT

A record of the current state of the art of simulation and the major part it now plays in policy formation in large organizations is provided by these conference proceedings. The 40 papers presented reveal an emphasis on the applications of simulation. In addition, the abstracts of 28 papers submitted to a more informal "paper fair" are also included. (RH)

FILMED FROM BEST AVAILABLE COPY

ED 077182

U.S. DEPARTMENT OF HEALTH.
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

1973
WINTER
SIMULATION
CONFERENCE

Edited By:
Austin Curwood Hoggatt
University of California

January 17–19, 1973

St. Francis Hotel
San Francisco, California

Sponsored by ACM/AIIE/IEEE/SHARE/SCi/TIMS

EM 011 010

CONFERENCE OFFICERS

GENERAL CHAIRMAN
Joseph Sussman
Massachusetts Institute of Technology
77 Massachusetts Avenue
Room 1 - 131
Cambridge, MA 02139
617 253-5320

PROGRAM CHAIRMAN
Austin Hoggatt
University of California
Center for Research in Management Science
26 Barrows Hall
Berkeley, CA 94720
415 642-4738

DEPUTY PROGRAM CHAIRMAN
Ernest Koenigsberg
Schools of Business Administration
University of California
Berkeley, CA 94720
415 642-1279

ARRANGEMENTS CHAIRMAN
Lawrence W. Heinle
Lockheed Palo Alto Research Laboratory
170 San Pablo Avenue
San Francisco, CA 94127
415 661-8555

TREASURER
David E. Watts
Bank of America
Management Sciences Dept. No. 3210
P.O. Box 37000
San Francisco, CA 94137
415 622-5223

REGISTRATION CHAIRMAN
Mary Jean Sortet
IBM
1501 California Avenue
Palo Alto, CA 94304
415 854-5538 X344

PUBLICATIONS CHAIRMAN
Robert D. Dickey
Golembe Associates, Inc.
One California Street, No. 2425
San Francisco, CA 94111
415 956-4010

REPRESENTATIVES OF SPONSORING ORGANIZATIONS

ACM
David H. Brandin

AIIE
Computer & Information Systems Division
Arnold Ockene

AIIE
Operations Research Division
A. Alan Pritsker

AIIE
Technical Division
William A. Smith, Jr.

IEEE
Computer Group
Harlow Freitag

IEEE
Systems Science and Cybernetics Group
Julian Reitman

SHARE
Stuart Trask

SCI
Jon N. Mangall

TIMS
Donald Heany

FROM THE GENERAL CHAIRMAN

The 1973 Winter Simulation Conference represents the sixth renewal of this annual conference and its first occurrence on the West Coast since December 1969. The conference has become a major focus of the discrete simulation professional community.

Traditionally, the conference has served as a forum for presentation of papers on the cutting edge of simulation application and methodology as well as a platform for tutorials by leaders in the field. This year is no exception. The program committee has succeeded in putting together an exceptionally strong technical offering.

I should emphasize that conferences like this one do not simply "happen". A great deal of work must be done by a variety of people. In addition to the writing and reviewing of papers and the organization of sessions, there is a great deal of administrative activity needed to coordinate an event of this size. I would appeal to you to get involved with our future conferences in a number of ways. I am sure that you will get even more out of the conference if you participate. Remember, the continued success of this series of meetings depends on you.

I hope that you enjoy the Conference and would personally welcome any suggestions that you might have. Next year, we'll all be in Washington, and I hope to see you all there.

Joseph Sussman
Cambridge
December 1972

CONTENTS

Panel 1: **Simulation's Role in Project Development** 1
Chairman: T. A. Marlow, Bechtel Corporation

Discrete - Event Simulation in the Engineering/Construction Industry 1
D. J. Dunne, Bechtel, Inc.

Experience with Simulation Models 1
T. A. Bratz, Bay Area Rapid Transit District

Session 1: **Behavior and Learning Models** 2
Chairman: G. Arthur Mihram, University of Pennsylvania

Towards a Simulation Model of Motivation and Adjustment 3
Paul F. Wyman, J. W. Slocum, Jr. and Richard Reed. The Pennsylvania State University

Simulation of an Individual Making Decisions Under Uncertainty 13
F. D. Tuggle, F. H. Barron and R. O. Day, The University of Kansas

Identification of Viable Biological Strategies for Pest Management by Simulation Studies 32
W. W. Menke, University of Florida

Session 2: **Simulation Methodology I** 51
Chairman: Mark Garman, University of California

Use of Simulation to Test the Validity and Sensitivity of an Analytical Model 52
Prosper Bernard

Multiple Sequence Random Number Generators 67
Joe H. Mize, Oklahoma State University

A 'Wait-Until' Algorithm for General Purpose Simulation Languages 77
Jean G. Vaucher, Universite de Montreal

Session 3: **Inventory and Distribution** 84
Chairman: Ernest Koenigsberg, University of California

Simulation of Sequential Production Systems with In-Process Inventory 85
David R. Anderson, Brain D. Sellers, Mohammed M. Shamma, University of Cincinnati

A Model for Analyzing Closed-Loop Conveyor Systems with Multiple Work Stations 93
Lynn E. Bussey, Kansas State University and M. Palmer Terrell, Oklahoma State Univerisity

A Cased Goods Conveyor Simulator 112
Donald A. Heimburger, The Procter and Gamble Company

A Simulator for Designing High-Rise Warehouse Systems 128
Kailash M. Bafna, Georgia Institute of Technology

Tutorial I: **GPSS** 139
Chairman: Thomas Schriber, University of Michigan

Session 4: **Health Services** 159
Chairman: Dean Uyeno, University of British Columbia

An Evaluation of Expanded Function Auxiliaries in General Dentistry 160
Kerry E. Kilpatrick, Richard S. Mackenzie, University of Florida

The Use of Computer Simulation in Health Care Planning 172
O. George Kennedy

A Simulation Model of a University Health Service Outpatient Clinic 199
Robert Baron, Edward Rising, University of Massachusetts

Session 5: **Simulation Methodology II** 226
Chairman: Michael Stonebraker, University of California

Constrained Sequential-Block Search in Simulation Experimentation 227
William E. Biles, University of Notre Dame

Optimization of Simulation Experiments 242
R. Taylor, W. Schmidt, V. Chachra, Virginia Polytechnic Institute and State University

A New Approach to Simulating Stable Stochastic Systems 264
Michael A. Crane, Donald L. Iglehart, Control Analysis Corporation

Session 6: **Manufacturing Applications** 273
Chairman: John W. O'Leary, Western Electric Company

Using an Extended Version of GERT to Simulate Priority and Assignment Rules in a Labor Limited Jobshop System 274
Michael J. Maggard, The University of Texas
William G. Lesso, The University of Texas
Bary L. Hogg, University of Illinois
Don T. Phillips, Purdue University

FROM THE PROGRAM CHAIRMAN

The Winter Simulation Conference has grown to be an impressive mixture of theory and application, languages and models, technicians and managers, and academic vs. government or industry. Simulation as a method of defining and solving problems is maturing in that major innovations have been made and the field is now exploiting footholds which have been gained in a number of applications areas. Thus we emphasize in this sixth annual conference the applications by which we test our shared approach to model building and analysis. The effort which has been put into this program has been amply rewarded and an overview of the simulation activity has emerged which may permit us to voice cautious optimissum about the future of the field. Simulation models are producing results; they are making significant contributions to knowledge and control of organizations. Read these proceedings-they are worthy of attention as a record of the current state of the art and practice of discrete event simulation and the major part it now plays in policy formation in large organizations.

Austin Curwood Hoggatt
Berkeley
December, 1972

Session 11: **APL Applications** 590
Chairman: C. Bartlett McGuire

Corporate Planning Model Design-Computerized Scratch Pads 591
Harley M. Courtney, The University of Texas

An Application of Simulation Models to Corporate Planning Processes 604
Ronald A. Seaberg, Xerox of Canada Limited

APL Models for Operational Planning of Shipment Routing, Loading and Scheduling 622
Richard D. Cuthbert, Xerox Corporation

A Two Asset Cash Flow Simulation Model 632
Richard C. Grinold and Robert M. Oliver, University of California

Session 12: **Gaming and Man-Machine Simulation** 641
Chairman: Richard Levitan, IBM Corporation

Progress Toward a Proposed Simulation Game Base for Curricula in Decision Sciences 642
Geoffrey Churchill and Edwin L. Heard, Georgia State University

Interactive Budgeting Models: A Simulation Tool for MIS Education 657
Theodore J. Mock, The University of California and Miklos Vasarhelyi, Pontificia Universdade Catolica do Rio de Janerio

The Traffic Police Management Training Game 677
Gay Doreen Serway, Allen S. Kennedy, Gustave Rath, Northwestern University

Tutorial 4: **Methodology for the System Sciences** 712
Chairman: G. Arthur Mirham, University of Pennsylvania

Evening 1: **Night in a Berkeley Laboratory** 715
Chairman: Austin Curwood Hogatt, University of California

A Micro-programmed APL Language Laboratory Control System
Austin Hoggatt, Mark Greenberg, University of California; Jeffrey Moore, Stanford University

A Business Game for Teaching and Research: Some Experiences 716
Martin Shubik, Yale University

The New York University Business Game 717
Myron Uretsky, New York University

Session 13: **Management Applications** 718
Chairman: Arnold Ockene, Securities Industry Automation Automatron Corporation

Corporate Simulation Models--A Reappraisal 719
William F. Hamilton, University of Pennsylvania

A General University Simulation Model 734
Charles C. Daniel, National Aeronautics and Space Administration and Hamal K. Eldrin, Oklahoma State University

Rate-Revenue-Cost of Service Simulation of a Natural Gas Utility 744
D. Jeffrey Blumenthal, On-Line Decisions, Inc.

A Simulated Investment Analysis for a Gas Pipeline Company 764
H. J. Miller, Colorado Interstate Gas Company

Session 14: **Languages for Simulation** 780
Chairman: Philip Kiviat, Federal ADP Simulation Center

An Interactive Simulation Programming System Which Converses in English 781
George E. Heidorn, Naval Postgraduates School

GASP IV: A Combined Continuous/Discrete FORTRAN Based Simulation Language 795
Nicholas R. Hurst and A. Alan B. Pritsker, Purdue University

NGPSS/6000: A New Implementation of GPSS 804
Karen Ast, Jerry Katzke, Jim Nickerson, Julian Reitman, United Aircraft Corporation; Lee Rogin, Naval Air Development Center

CMS/1 - A Corporate Modeling System 814
R. F. Zant, Clemson University

Session 15: **Maintenance and Reliability** 828
Chairman: Richard E. Barlow, University of California

A Reliability Model Using Markov Chains for the Utility Evaluation of Computer Systems Onboard Ships 829
Carsten Boe, Tor Heimly and Tor-Chr. Mathiesen, Det Norske Veritas

Monte Carlo Simulation of Crosstalk in Communication Cables 844
Aridaman K. Jain, Bell Telephone Laboratories, Inc.

Incorporation of False Alarms in Simulations of Electronic Receivers 858
V. P. So[illegible]zynski and C. J. Pearson, SYCOM, Inc.

Tutorial 5: **SIMSCRIPT** 868
Chairman: F. Paul Wyman, Pennsylvania State University

Evaluating Job Shop Simulation Results 289
Carter L. Franklin, University of Pennsylvania

Simulation Applied to a Manufacturing Expansion Problem Area 298
J. Douglas DeMaire, Olin Corporation

Cycle-Time Simulation for a Multiproduct Manufacturing Facility 311
M. M. Patel, J. M. Panchal, and M. T. Coughlin, IBM Corporation

Panel 2: **Simulation in Government** 319
Chairman: Peter House, Office of Research EPA

Session 7: **Urban Problems** 319
Chairman: Gary Brewer, The RAND Corporation

A Demographic Simulation Model for Health Care, Education, and Urban Systems Planning 320
Philip F. Schweizer, Westinghouse Electric Corporation

Simulation Model of New York City's Felony Adjudicatory System 334
Lucius J. Riccio, Lehigh University

A Simulation Model of the New York City Fire Department: Its Use as a Deployment Tool 353
Grace Carter, Edward Ignall, Warren Walker, The New York City RAND Institute

On-Line Simulation of Urban Police Patrol and Dispatching 371
Richard C. Larson, Massachusetts Institute of Technology

Session 8: **Aerospace Applications** 386
Chairman: Lawrence W. Heinle, Lockheed Palo Alto Research Laboratory

Digital Simulation of a Multiple Element Threat Environment 387
K. E. Dominiak, University of Florida and R. J. Ireland, Honeywell Information Systems

Cost/Resource Model 396
Betty J. Lanstra, Hughes Aircraft Company

MAFLOS--A Generalized Manufacturing System Simulation 416
K. Mitome, S. Tsuhara, S. Seki, K. Isoda, Hitachi, Ltd.

A Description of an AAW Model and Its Classroom Uses 426
Alvin F. Andrus, Naval Postgraduate School

Simulation in the Design of Automated Air Traffic Control Functions 449
Paul D. Flanagan, Judith B. Currier, Kenneth E. Willis, Metis Corporation

Session 9: **Financial Models (General)** 463
Chairman: Theodore Mock, University of California

Accounting Rate of Return vs. True Rate of Return: Considering Variability and Uncertainty 464
John V. Baumler, Ohio State University

Variability Assumptions and Their Effect on Capital Investment Risk 481
F. J. Brewerton, Louisiana Tech University, W. B. Allen, United States Air Force

A Computerized Interactive Financial Forecasting System 497
Philip M. Wolfe and Donald F. Deutsch, Motorola Inc.

Multinational Capital Budgeting: A Simulation Model 512
Thomas J. Hindelang and Andre Fourcans, Indiana University

Tutorial 2: **GASP** 535
Chairman: A. Allen B. Pritsker, Purdue University

Tutorial 3: **Simulation of Econometric Models** 535
Chairman: B. F. Roberts, University of California

Session 10: **Transportation Models** 536
Chairman: Richard de Neufville, Massachusetts Institute of Technology

Simulation Analysis of Marine Terminal Investments 537
David W. Graff, Esso Mathematics & Systems, Inc.

Simulation in the Design of Unit Carrier Materials Handling Systems 546
W. Wayne Siesennop, University of Wisconsin; Fritz A. Callies, Rex Chainbelt, Inc.; Neal S. Campbell, A. O. Smith Corporation

A Generalized Model for Simulating Commodity Movements by Ship 567
John C. Rea, Pennsylvania State University; David C. Nowading, The University of Tennessee; Philip W. Buckholts, R. Shriver Associates

Simulation of Garland, Texas Vehicular Traffic Using Current and Computed Optical Traffic Settings 580
Frank P. Testa and Mark Handelman, IBM Corporation

PAPER FAIRS* 869

The Daughter of Celia, the French Flag, and the Firing Squad 870
G. T. Herman, W. H. Liu, State University of New York at Buffalo

An Application of Simulation to Debugging and Maintaining a Computer Network System 871
M. W. Collins, D. G. Harder, R. C. Jones, University of California

An Experimental Evaluation of Monte Carlo Simulation in MIS Project Decisions 872
John B. Wallace, Jr., University of Florida

Robustness and Analytic Computer System Models 873
W. R. Franta and R. Vavra, University of Minnesota

A Model for Simulating and Evaluating the Response of a Management Information System 874
Hamed K. Eldin, David Shipman, National Aeronautics and Space Agency

Trace Driven System Modeling 875
J. F. Grant, IBM Corporation

ARPEGE: Simulation of an Air Pollution Crisis 876
M. Greene, A. Hockhauser, M. Reilly, A. Walters, Carnegie-Mellon University

A Computer Systems Design Game 877
Norman R. Lyons, Cornell University

A Deterministic Simulation Model for Scheduled Airline Fleet Maintenance 877
Alan J. Parker, Florida International University

ASSET: A Digital Computer Language for the Simulation of Communication Systems 878
R. R. Bowen, R. V. Baser, C. D. Shepard

Simulation Model to Evaluate the Role of a Multiphasic Screening Unit in the Health Care Delivery System 879
F. Delaney, M. Oppenheim, M. Goldberg, W. Schumer, A. Greenburg

Medical Care Simulation: A Study Utilizing Dynamic Simulation Modeling 880
S. H. Cohn and J. F. Brandejs, University of Toronto

Simulating the Impact of Expanded Deligation of Dental Procedures 881
J. B. Dilworth, W. J. Pelton, O. H. Embry, G. A. Overstreet, The University of Alabama

A PL/1 Model of an Emergency Medical System 882
Kenneth F. Siler, University of California

Simulation of an Epidemic: Development of Control Strategies of Schistosomiasis 883
Ken-Lon Lee

Simulation Analysis of an Emergency Medical Facility 884
E. C. Garcia, W. F. Hamilton, J. W. Thomas, University of Pennsylvania

An Interactive Multi-Item Inventory Computer Simulation Model 884
Wayne Shiveley, Lehigh University

GWSS - A Generalized Warehouse Simulator System 885
Alvin M. Silver, Dasol Corporation

A Dynamic Control System for Hospital Inventories 886
James D. Durham, Stephen D. Roberts, Medicus Systems Corporation

Distribution Combining Program 886
Oldrich A. Vasicek, Wells Fargo Bank

MATHRISK 887
Stephen L. Robinson, Mathematica, Inc.

MATHNET 888
Stephen L. Robinson, Mathematica, Inc.

A Risk-Return Simulation Model of Commodity Market Hedging Strategies 889
Robert E. Markland, Robert J. Newett, University of Missouri

Using the Computer to Plan Production in a Flow Shop 890
Dana B. Hopkins, Jr., Babcock and Wilcox

A Simulation Study of Basic Oxygen Furnace Operations 891
C. Jain, Cleveland State University

Autonetics Planned Production Line Evaluation Simulator 892
M. P. J. Moore, North American Rockwell

A Performance Evaluation Technique for the Measurement of a Facility's Ability to Process the Proper Jobs 893
J. J. Babiel and B. Z. Duhl, IBM Corporation

A Directed Approach to Selecting a Sequencing Rule 894
J. C. Hershauer, R. J. Ebert, Arizona State University

KEY WORD INDEX 895

PANEL 1: SIMULATION'S ROLE IN PROJECT DEVELOPMENT

Chairman: Thomas A. Marlow, Bechtel, Inc.

This panel will discuss some of the factors which bear on the problem of placing simulation in its proper perspective in a total project development effort. The process of how management evaluates whether or not simulation is a viable management aid to the decision-maker involved in some specific project development programs will be addressed. Several examples will be given and more than one simulation approach will be covered. Examples will be taken from both the public and private sectors.

DISCRETE-EVENT SIMULATION IN THE ENGINEERING/CONSTRUCTION INDUSTRY

D. J. Dunne
Bechtel, Inc.
San Francisco, California

The conceptual and construction procedure planning process for modern major resource development projects represents a challenging problem in scheduling, logistics and resource allocation. The large capital investment, the magnitude of many of the projects, the impact of minor decisions, and the fact that many resource extractions are located in remote regions of the world combine to present a potentially high degree of risk to management. This paper discusses why discrete systems simulation can be a valuable planning tool to aid management in preparing for the real world contingencies which inevitably occur when major projects move into the field.

"Experience with Simulation Models"

T.A. Bratz, Bay Area Rapid Transit District

Mr. Bratz will comment on two simulators which have been developed and used by Bay Area Rapid Transit District, one which was created for the purpose of designing the system and the other which was created to test the design.

Session 1: Behavioral and Learning Models
Chairman: G. Arthur Mihram, University of Pennsylvania

Currently, the most challenging aspect of simular methodology is the representation of the behavior and decision processes in animals, including man. One approach, especially suited for miming human decision-making, is the construction of gaming models, in which selected humans participate as physical entities at pertinent points in simular time.

The construction of pertinent algorithms, meaningful to simulations which mime behavioral and learning processes, constitutes an important inter-disciplinary topic of the contemporary systemic sciences, and involves psychologists, neurologists, statisticians, and systemic scientists who construct credible simulation models.

The Behavioral and Learning Models Session includes: two papers which stress the important role that simulation modeling is playing in the development of our understanding of these two areas; than, a panel discussion whose members shall address the need for the application of both stringent verification tests and assiduous validation tests in order to establish credible simulation models; and, a final paper which indicates the role, which an understanding of human behavioral and learning processes must occupy, in establishing credible simulation models of environmental and societal systems.

Papers

"A Computer Simulation of Maslow's Need Theory"
Paul F. Wyman, Pennsylvania State University; John W. Slocum, Jr., Pennsylvania State University; Richard Reed, Electronic Data System

"Simulation of an Individual Making Decisions Under Uncertainty"
F. D. Tuggle, F. H. Barron, and R. O. Day, University of Kansas

"Identification of Viable Biological Strategies for Pest Management by Simulation Studies"
W. W. Menke, University of Florida

Panel Discussants

Daniel N. Braunstein, Oakland University
John V. Dutton, New York University
Austin Hoggatt, University of California
John F. Lubin, University of Pennsylvania
Martin Shubik, Yale University

TOWARDS A SIMULATION MODEL
OF MOTIVATION AND ADJUSTMENT

F. Paul Wyman
John W. Slocum, Jr.
Richard R. Reed

The Pennsylvania State University

Abstract

A model of human motivation is formulated on the basis of Maslow's need theory. Additionally, behaviors are selected by degree of tension and are reinforced by environmental reactions which facilitate or frustrate reduction of tension according to the aggressiveness of behavior. Comparisons of alternative environments support the internal consistency of the model. Recommendations are made to improve the stability of the model and for behavioral research that would be necessary for validation.

INTRODUCTION

During the past several years, computer simulations of personality have been developed. The personality theorist is concerned with attempting to identify and classify similarities and differences between people. But it is not merely transitory similarities and differences among people that intrigue the personality theorist. The data he wishes to interprete and understand are abstracted from characteristics showing continuity over a period of time. The important characteristics which he focuses upon seem to have psychological importance to the individual, such as feelings, needs, actions, leaving to the biological scientist such continuous aspects of functioning as acetylcholine cycles and blood pressure.

All of these simulation efforts are laudable as long as there is a possibility of testing the predictions of a personality theorist's model against reality. If the model is untestable, then it is a futile exercise to simulate it and then to attempt a proper validation. For example, what evidence is offered by Freud or his disciples that the basic variables--id, ego and superego--postulated by them exist? Do people truly pass through the oral, anal, phallic, and gential phases of development? Unfortunately, most of the empirical data of psychoanalysts are available in the form of psychotherapy and not as results of statistically analyzed nor operationalized constructs. For these reasons the verification of computer simulations of personality are subject not only to the limitations of statistical analysis but even more to the problem of getting data for validation.

Recognizing these limitations we feel that the process of systematic modeling, such as one must perform in computer simulation, contributes to the operationality of vaguely-defined theories. By exposing a tentative set of intuitive mechanisms to open criticism, perhaps we can move closer to a fuller understanding of personality theories. It was also our intention to show the utility of a computer simulation as a vehicle for integrating theoretical concepts of motivation and adjustment. Therefore our simulation is limited here to validation utilizing merely potentially operational constructs. The data gathered from the simulation may be evaluated only at the level of subjective comparison from obtainable reports and other research findings.

One of the determinants of an individual's personality is a need. A need, drive, motive or habit are alternative ways of conceptualizing subunits of personality whose interplay define the course of an individual's action and development. A need then may be capable of accounting for a variety of behavior. In most of the computer simulations of personality (Loehlin, 1968; Dutton and Starbuck, 1971; Tompkins and Messick, 1963; Colby, 1964 and Moser, et al., 1970), motivational constructs have not been emphasized because these models are not strongly oriented toward action. It is the purpose of this simulation to model Maslow's theory of motivation which emphasizes the behavior of the individual.

The essence of Maslow's theory of motivation (1970) is presented below. According to Maslow (1943), each individual strives to actualize (grow) and avoid deprivation. Deprivation motivation refers to the urge to strive for the goal states, presently unachieved, that are necessary in order to ease the pain and discomfort due to their absence. The aim of deprivation motivation is to decrease the organismic tension buildup through deficit states that represent deviations from homeostatic balance. Actualization has to do with the realization of capabilities or ideals. As the individual strives to realize these ideals, he engages in more complex differentiated behavior that may increase his level of tension. Maslow's theory requires that a goal state be achieved and that the individual will engage in behavior that is instrumental to reaching the goal. The goal states have been defined in terms of five basic needs. These needs, arranged in a hierarchy of ascending order, are: physiological, safety, love, esteem, and self-actualization. This classification of needs is interpreted that if an individual's physiological needs are unsatisfied, he will take action to alleviate this unsatisfied condition because an unsatisfied need introduces tension within the individual which he is trying to avoid and/or reduce (Maslow, 1970). If the need remains unsatisfied for a period of time, the level of tension accumulates within the individual until it reaches some threshold which forces the individual to behave in a manner alleviating the tension. As indicated in Figure 1, behavior can be either passive or aggressive. Passive behavior may take the form of withdrawal from the tension arousing situation and is manifested by such phenomenon as absenteeism, turnover and the like (Vroom, 1964; Fournet, DiStefano and Pryer, 1966, and Lawler, 1970). Aggression is most readily observable as a move outward which attacks the source of tension. Aggressive behavior, such as wild cat strikes, grievances and the like are industrial examples of this type of behavior engaged in by individuals to alleviate tension (Lawler, 1970). But behavior is not necessarily confined to dysfunctional acts. The individual who cannot satisfy his need may engage in constructive search-directed behavior that will attempt to satisfy the need. Empirical evidence suggests that limited deprivation of need satisfaction can lead to improved decision-making processes, greater motivational force, and the like (Kolasa, 1969).

As for the emergence of a new need after the satisfaction of the most prepotent need, this emergence is not sudden but rather gradual, occurring by slow degrees from nothingness to some level of tension. This new need then directs the person's behavior until it has been reduced to a "satisfactory" tension level (Slocum, 1971).

PROBLEM SITUATION

The purpose of this simulation is to represent an individual experiencing needs, being motivated by a need toward a behavior, and having that behavior and an environment influence the level of needs. The fundamental notion to be examined is the variation in the need levels of an individual and the change in his degree of aggressiveness toward a given environment. The flow chart in Figure 1 shows the various components and inter-relations of components included in the model.

To develop a simulation model to explain Maslow's theory, in SIMSCRIPT II, we defined needs as entities with attributes of tension. Each need enters a conscious state at some threshold level. Tensions are held to grow exponentially over time. Tension release can modify the threshold level of the need, thus, providing a feedback effect. Needs are selected for action on the basis of the greatest tension. Each of several possible behaviors has a perceived degree of justifiable aggressiveness. The most aggressive action is merited by the total motivational force. Tensions and threshold level are modified by environmental reactions, which are favorable if the action corresponds to what society sees as justifiable aggressiveness. If an action successfully reduces tension below the threshold, the behavior is reinforced. Thus, an operant conditioning feature is built into the model. The variables studied include the time path of tensions as well as an aggregate measure of the perceived justifiable degree of aggressiveness. These time series are evaluated for equilibrium and degree of variability. The model is programmed in SIMSCRIPT II using entity-set features with a fixed-time increment time flow mechanism (Kiviat et al., 1968).

MODEL DETAILS

Propensity for Action

Each need has an associated tension which, if unfulfilled, increases exponentially over time (Maslow, 1970). The tension of the i^{th} need is modeled as follows:

$$\text{Tension}_i = a_i \exp(b_i t) \qquad (1)$$

where a_i denotes the initial level (at time t=0) of Tension_i and b_i denotes the rate of tension increase. The amount of change in tension is computed as the derivative of Eq.(1) times a fixed time increment Δt. The total tension is then a summation of all individual tensions.

Threshold and Sensitivity to Tension

For values of Tension_i less than Threshold_i, the individual may be aware that Tension_i exists but its magnitude is such that the individual ignores it. Upon exceeding Threshold_i, however, the individual experiences such discomfort that he must recognize Tension_i and take action to reduce it. An increase in sensitivity to tension is modeled by lowering the value of Threshold_i. If an individual has been successful in satisfying Need_i then he will be less sensitive to Tension_i in the future. This is represented in the model by raising the value of Threshold_i.

The Fundamental Cycle

The propensity to act on Need_i as a result of Tension_i is given by:

$$P_i = 1.0 - (\text{Threshold}_i/\text{Tension}_i) \qquad (2)$$

Tension_i and Threshold_i are initialized at values of 1.0. As time increases by Δt, successive values of P_i are calculated for successive values of Tension_i. As each Tension_i exceeds its particular Threshold_i, it will be recognized and compared to all tensions that have exceeded their threshold value by examining only the needs with P_i's>0. The need with the largest P_i (the greatest recognized tension) will motivate action. If two or more needs have the same tension, then the most potent need is determined by Maslow's need hierarchy.

As a result of the given action, selected Tension_i's will be positively or negatively changed representing the reaction of the environment to the action. In addition, b_i the rate of change of Tension_i may be changed. Success in eliminating tension will result in an increased tendency to adopt that behavior in the future.

We assume that each action requires a minimum of one time increment (1 hour) to perform. At each increment of time, the values of all P_i's are re-evaluated to determine which need will be responded to next. If no need demands attention, then time continues to be incremented.

If an action requires more than one time increment then the relevant Need_i is removed from the "set" of needs requiring attention until the action is completed. Tension for the removed need decreases at each time increment.

The Determination of Behavior

The individual's motivation to undertake a particular action is related to the amount of tension the individual experiences because of his most potent need (P_i') and his general state of tension (ST). The propensity to act, P_i' of equation (2) is combined with the state of tension (ST) to represent the total motivating force of Need_i. The total motivating force, TMF_i, is conceived as the sum of P_i and ST, i.e.,

$$\text{TMF}_i = P_i' + \text{ST} \qquad (3)$$

where P_i' = current P_i of the selected need

and $\text{ST} = \Sigma_i \overline{P}_i$

where $\overline{P}_i$ = time averaged value of past P_i's

For each need there is a unique set of actions from which one action is chosen to satisfy the need. An action is selected using the value of TMF_i (3). Every action has an attribute X_{ij} which symbolizes the individual's perception of the TMF_i required to justify the aggressiveness of the $j^{\underline{th}}$ action to satisfy the $i^{\underline{th}}$ need. The X_{ij} values have been assigned so that larger numbers correspond to more aggressive acts. In our example we let the X_{ij} values be 0.1, 0.2, 0.3, 0.5, 0.7, and 0.9 for the six acts of each need. That act will be chosen whose X_{ij} is just larger than the TMF_i value.

In principle, the X_{ij}'s could be determined empirically by asking subjects to rank order a set of specific behavior descriptions. By anchoring weights of extreme behavioral incidents, "perceived justifiable aggressiveness could be operationalized.

The Environmental Reaction

This environment in which the individual is placed provides a matrix of approving and disapproving forces for every action he may take. If society approves a behavior, then tension is more effectively reduced than if society disapproves a behavior. Each behavior can either increase tension or reduce tension depending on the environmental reaction. A behavior has direct impact on the need toward which it is oriented, but it may have side effects on the tensions of other needs. For instance, if a person is exceedingly hungry, he may steal to satisfy his hunger, but in so doing his environment may cause his need for safety and self-esteem to increase.

The $j^{\underline{th}}$ action has associated with it a set of environmental factors, E_{ij} (one per need) which modify the corresponding Tension_i. To represent the environmental reaction, Tension_i's are multiplied by the corresponding E_{ij} of the selected $j^{\underline{th}}$ action. These E_{ij} factors range in scale from 0.25 to 1.75.

The E_{ij} could be measured, in principle, by a panel of qualified judges who rank the tension-reducing or increasing effect that the environment (both physical and social) would exert in response to a set of specified behaviors. Again, these effects could be rank ordered and related to a set of anchored critical incidents to obtain estimates of the E_{ij}'s for a given set of actions.

The environmental reaction is defined as being critical, non-critical, or neutral. If the environment is critical, then it will change tension values by a larger amount than if it were non-critical. Hence, there are five possible reactions:

1. Critical Reduction
2. Non-critical Reduction
3. Neutral
4. Critical Increase
5. Non-critical Increase

For the purpose of this simulation, a given action will provoke either all critical reactions or all non-critical reactions. This implies the environment will react consistently to any given action. Table 1 shows the values assigned to each environmental reaction.

REINFORCEMENT OF BEHAVIOR

In each time cycle the P_i are first computed. Next a "revised" Threshold_i is calculated. For the need selected for action, tension may be reduced below the revised Threshold_i. If this is the case, the Threshold_i is increased to reflect reduced sensitivity to this tension in the future. If Tension_i remains above the Threshold_i, then the Threshold_i is carried into the next time cycle unmodified. For the needs not selected for action, the "revised" threshold is also carried forward into the next time cycle hence the Threshold_i's of these needs are being lowered thus reflecting to tension in the future.

For the need singled out for action, the behavior may be successful in reducing Tension_i below Threshold_i as noted above. If this occurs then the appropriate X_{ij} is reduced by .001, so that less total motivational force TMF_i will be necessary to select a behavior of similar aggressiveness in the future.

PROBLEM STATEMENT

One of the important dependent variables of this model is BI_i the value of the behavioral index for Need_i. For each need, the BI_i value is equal to the average value of X_{ij}'s of those actions which can be selected to satisfy Need_i.

For the purpose of this paper, questions regarding the following cause and effect relationships will be posed:

1. Given initial values for the propensities to act, P_i's, behavior index, BI_i's, and state of tension, ST, what is the effect of varying patterns of environmental responsiveness on the propensities to act (P_i's)?
2. Given the same initial conditions, what is the effect on the behavior aggressiveness index, BI_i, for varying patterns of environmental responsiveness?
3. Given the same initial conditions, what is the effect on the state of tension, ST, for varying patterns of environmental responsiveness?

For each pattern of environmental responsiveness a time series of values was generated for P_i, ST, and BI_i. A tabular comparison was made showing the differences in the time series as a result of the four different environments.

EXPERIMENTAL DESIGN AND RESULTS OF VERIFICATION

The environmental "patterns" take on four levels. Each level depicts two values the EW's one for increasing tensions and a second for decreasing tensions. Table 2 gives these values with the starting values for the propensities to act, behavior index (dispositions for aggressiveness), and state of tension, ST.

The environments have been chosen to be either critical both in reducing and increasing tension or non-critical both in reducing and increasing tension. The initial values for the propensity to act are determined from tension values equal to the threshold values. The initial behavior index is defined as the average of initial X_{ij}'s (justifiable aggressiveness) equal to 0.1, 0.2, 0.3, 0.5, 0.7, 0.9 for each need.

Table 3 depicts the results of our experimental runs. For those environments tested, the values of all Tension_i's became very large. Corresponding to this increase, there is also an increase in the state of tension. As the propensity to act and the state of tension increase, the behavior index tends to decrease, indicating that the disposition for aggressiveness increases directly with tension accumulation, as expected.

The results indicate that further experiments will have to be run to determine if any environmental factors exist which will permit a stabilization of tension. It is quite possible that the exponential function of time is not appropriate for this type of simulation and that such a function prevents stability after long periods of time. This is suspected because for large values of time, the growth of tension becomes very rapid.

CONCLUSIONS AND RECOMMENDATIONS

The basic model provides results that are consistent with the fundamental expectations of the simulation. The behavior index decreases for increasing values of tension. The point in time at which the Tension_i's become unstable is prolonged for an environment which causes a larger reduction in tension. Also the increase in the state of tension for all experiments depicts lack of success in satisfying needs, not an uncommon situation.

Our present results indicate that the model becomes unstable with time. It is felt that the function for tension growth requires modification in this simulation. It is suggested that the exponential function of time is not appropriate or that the value of the exponent in Equation 1, should be modified. It is recommended that such an investigation be conducted to determine a time function which does provide stability.

A great deal of benefit of simulation results from the modeling process itself, rather than solely from numeric results of the simulation. We made several assumptions in the course of our modeling process to provide necessary constructs for integrating the concept of motivation, learning, and socialization theory. We now recommend investigation of the following assumptions by psychological and behavioral research to evaluate the merit of our model's structure: the existence of a tension threshold; the notion of a changeable "threshold"; effects of successful action on perceived justifiable aggressiveness;

the summative effect of individual tensions upon motivation; and the exponential rate of growth of tension resulting from need deprivation. While the model is admittedly embryonic in its state of development, we feel the exercise has been worthwhile for the benefits of conceptual integration that it has stimulated.

REFERENCES

1. Colby, K. Experimental Treatment of Neurotic Computer Programs. Archives of General Psychiatry, 1964, 10, 220-227.
2. Dutton, S. M. and Starbuck, W. H., Computer Simulation of Human Behavior, John Wiley and Sons, Inc., New York, New York, 1971.
3. Fournet, F., DiStefano, M., and Pryer, M. Job Satisfaction: Issues and Problems. Personnel Psychology, 1966, 19, 165-183.
4. Kiviat, P. J., Villanueva, R., and Markowitz, H. M., The SIMSCRIPT II Programming Language, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1968.
5. Kolasa, B. Introduction to Behavioral Science for Business. New York: John Wiley and Sons, 1969, 241-278.
6. Lawler, III, E. Job Attitudes and Employee Motivation: Theory, Research and Practice. Personnel Psychology, 1970, 23, 223-237.
7. Loehlin, J. C., Computer Models of Personality, Random House, New York, New York, 1968.
8. Maslow, A. H., Motivation and Personality, 2nd ed., Harper & Row, New York, New York, 1970.
9. Maslow, A. H., A Theory of Human Motivation, Psychological Review, 1943, 50, 370-396.
10. Moser, U., Von Zepplein, and Schneider, W., Computer Simulation of a Model of Neurotic Defense Processes, Behavioral Science, 1970, 15, 194-202.
11. Slocum, J. Motivation in Managerial Levels: Relationship of Need Satisfaction to Job Performance. Journal of Applied Psychology, 1971, 55, 312-316.
12. Tomkins, S. S. and Messick, S., Computer Simulation of Personality, John Wiley and Sons, Inc., New York, New York, 1963.
13. Vroom, V. Work and Motivation, New York: John Wiley and Sons, 1964.

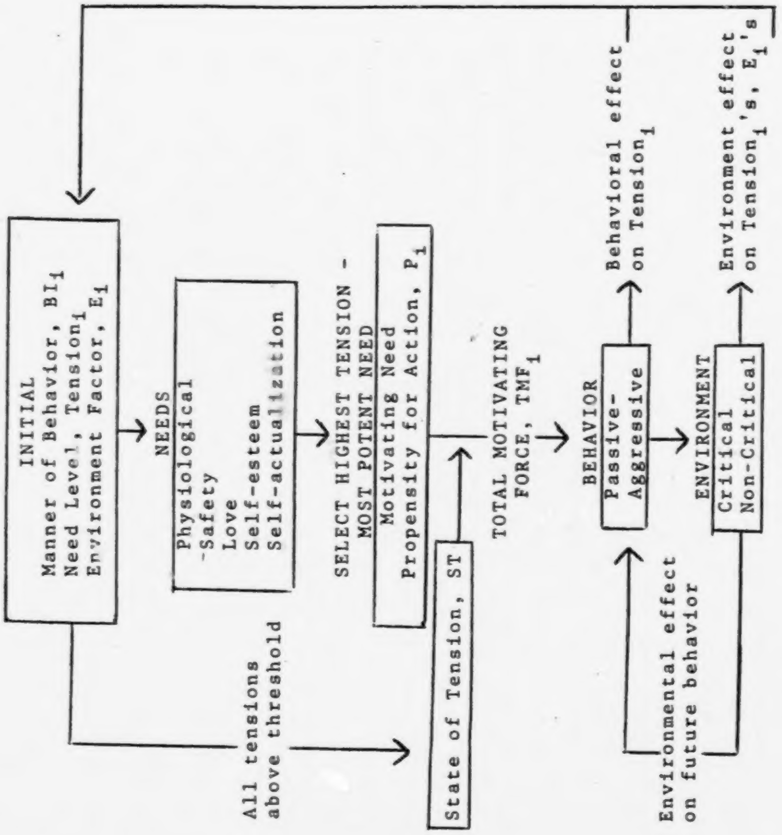

INITIAL
Manner of Behavior, BI_i
Need Level, $Tension_i$
Environment Factor, E_i
NEEDS
Physiological
-Safety
Love
Self-esteem
Self-actualization
SELECT HIGHEST TENSION - MOST POTENT NEED
Motivating Need
Propensity for Action, P_i
All tensions above threshold
State of Tension, ST
TOTAL MOTIVATING FORCE, TMF_i
BEHAVIOR
Passive-Aggressive
Behavioral effect on $Tension_i$
Environmental effect on future behavior
ENVIRONMENT
Critical
Non-Critical
Environment effect on $Tension_i$'s, E_i's

Figure 1
Flow Chart of Model

	Value of Tension Reducing Factors	Value of Tension Increasing Factors
Critical Environment	0.25	1.75
Non-critical Environment	0.75	1.25
Neutral	1.00	1.00

Table 1 - Environmental Factors, E_i

	EXPERIMENTS			
Initial Values	1	2	3	4
Propensity to act for all needs	0.0	0.0	0.0	0.0
Behavior index for all needs	0.45	0.45	0.45	0.45
State of Tension	0.0	0.0	0.0	0.0
Environment factors (average E_i's)	Non-critical	Critical	Non-critical	Critical
Increasing	1.0	1.0	1.25	1.75
Decreasing	0.75	0.25	0.75	0.25

Table 2 - Experiments for Simulation

Table 3
Effect of Environmental Reaction
on Simulation Time Series

Environmental Effect on Increasing Tension	Reducing Tension											
	Critical Environment						Non-Critical Environment					
	Neutral			Critical			Neutral			Non-Critical		
Simulated Time (Hrs.)	P_1*	ST*	BI*	P_1	ST	BI	P_1	ST	BI	P_1	ST	BI
300	0**	0	.48	.001	0	.50	0	0	.45	.005	0	.50
400	0	0	.46	.003	0	.48	0	0	.40	.006	0	.42
500	0	0	.47	.005	0	.42	.002	0	.40	.015	.002	.35
600	.006	0	.40	.010	0	.40	.100	0	.39	.090	.005	.32
700	.018	0	.39	.020	.002	.39	.250	.005	.37	1.00	.050	.30
800	.060	.005	.32	.110	.005	.33	.800	.040	.33	***		
900	.500	.020	.30	1.00	.020	.33	.900	.100	.31			
1000	1.00	.100	.27	***			1.00	.200	.28			
1100	1.00	.180	.21				1.00	.250	.28			
1200	1.00	.200	.20				1.00	.270	.30			

*The following explains the series symbols:
P_1 = Propensity to Act
ST = State of Tension
BI = Behavior Index
(all have the range 0 to 1)

** All entries with "0" indicate values are less than .001.

*** All entries left blank are due to termination of the simulation at these points.

SIMULATION OF AN INDIVIDUAL MAKING DECISIONS UNDER UNCERTAINTY

Francis D. Tuggle,
School of Business
and
Department of Computer Science

F. Hutton Barron
School of Business

Richard O. Day
Department of
Computer Science

University of Kansas
Lawrence, Kansas 66044

Abstract

A computer simulation model, SIDIP (Simulation of Individual Decisions through Information Processing), of a person making nine decisions under uncertainty is sketched. Eight of a subject's (S's) choices are consistent with the Laplace or maximize expected value criteria and S's other is consistent with the Savage (minimax regret) criterion (see Luce and Raiffa, 1957). SIDIP implies that the subject does not use the conventional computational processes dictated by those criteria. SIDIP reproduces S's articulated choice behavior: inconsistent use of choice criteria, rejection of some alternatives, and eventual choice from the preferred alternatives. Analysis of information processing models of suboptimal decision behavior suggests operational techniques by which decision making can be improved.

I. Introduction

A. Setting

The problems of individual decision making have been classified by Luce and Raiffa (1957) as decision making under (1) certainty, (2) risk, (3) uncertainty, and (4) partial ignorance (a combination of risk and uncertainty). These four classes of decision problems are defined in terms of knowledge of the probability distribution over the states of nature, given the usual decision theoretic formulation (decision maker; choices, acts, or alternatives; states of nature; payoffs) of a decision situation. Thus, decision making under certainty is trivial from a decision theoretic point of view.

Normative decision theory prescribes choice for a given structure and classification (risk, uncertainty, or partial ignorance) by specifying a criterion of choice. Normative theory prescribes in the sense that, if the criterion and formulation are accepted, choice is unambiguous. There are several decision theories -- meaning that for certain conditions there are several "reasonable" choice criteria.

It is well known that people do not always behave in a manner consistent with various normative decision theories. The descriptive failures of normative criteria are documented elsewhere (e.g., under risk, MacCrimmon, 1969; under uncertainty, Tuggle, 1972; under partial ignorance, Barron, 1970). Since these experimental results were derived from laboratory studies using reasonably artificial problems, it is likely that actual decisions made daily by decision makers facing complex real-world problems would also exhibit inconsistencies with normative theory.

In this paper we choose to study decision making under uncertainty. We believe partial ignorance is a reasonable representation of real world decision problems; however, several proposed approaches for dealing with partial ignorance first reformulate the problem as decision making under uncertainty. (Those who subscribe to a subjective or personalistic theory of probability would convert partial ignorance to risk. We temporarily reject risk since in many problems the probabilities are, at best, vaguely known and the decision maker is unwilling to accept the probability estimates for decision making purposes.) Other possible approaches to partial ignorance include deciding "as if" it were an uncertainty situation or deciding "as if" it were risk (i.e., maximize expected value or expected utility), but first rejecting (or considering) alternatives based on uncertainty criteria. These approaches place heavy emphasis on decision-making under uncertainty.

B. Proposal

Our aim is to study individual human decision-making under uncertainty so as to learn how decision processes are used and how to introduce realistic modifications into a person's cognitive behavioral repertoire so that he makes an optimal decision. This paper puts heaviest weight on unravelling and simulating nonoptimal decision processes; later papers will address the second subgoal of internalizing different cognitive processes.

In order to ensure that our understanding of current (suboptimal) human decision-making processes is explicit, operational, and falsifiable, we have encoded our model as a computer simulation program. Our program, entitled SIDIP for Simulation of Individual Decision-making through Information Processing, is described in detail in Section IV and is tested and analyzed in Section V.

In order to have a framework in which to express our model, we chose the Information Processing System (IPS) approach of Newell and Simon (1972). Accordingly, our mode of operation is as follows: first, we take verbal protocols from a subject while he is making decisions (see Sections II and III). Second, we construct an IPS model of the subject's cognitive processes (see Section IV). Third, we examine, both qualitatively and empirically, the adequacy of the IPS model (see Section V). Fourth, we use an IPS model assumed to be validated and from it infer reasons why the subject did not comply with the set of normative decision processes (see Section VI, Part B). Fifth, knowing the subject's IPS, we suggest changes in his information processing to get conformity to normative theory (Section VI, Part B); and last, we suggest how actually to implement the modified processes (Section VI, Part B).

II. Method

A single subject (S) was enjoined to make decisions under uncertainty and to verbalize as much of his thought process as possible. His utterances were recorded on audio tape and later transcribed to paper. S's protocol is analyzed in the next section. A computer program (SIDIP) was written (see Section IV) to simulate the essential parts of S's decision-making behavior. Goodness-of-fit tests of SIDIP's behavior to S's behavior are performed in Section V. Action recommendations are made in Section VI.

The subject was faced by nine decision situations, which were sequentially presented to him. The nine decision situations are independent; S was not permitted to see the next situation until he had made a final choice on the previous one. After all nine choices had been made, a single situation was selected by the experimenter (E) to be played for real money. The entire session with S lasted about 50 minutes.

A. Decision Problems

The nine situations faced by S were tabulated as nine different payoff matrices for uncertain decisions. Table 1 exhibits the first matrix that S was given. (Copies of all instruments used in this work are available from the authors.) Each matrix contains eight rows, corresponding to the strategies or actions available to S, who had to select one of them. The four columns correspond to the possible states of nature that could occur. S did not know what process was to be used for generating the states of nature: uniform random, friendly (maximax), antagonistic (minimax), or some other process.

		States			
		ZEJ	XEQ	WUH	QUG
Decision Strategies	S_1	12	0	4	4
	S_2	2	7	6	5
	S_3	0	11	3	4
	S_4	4	6	6	4
	S_5	10	4	4	2
	S_6	4	5	4	5
	S_7	4	9	2	3
	S_8	8	2	6	6

Table 1: First Decision Situation

To prevent learning S was never told what state of nature occurred after his row choice.

Each of the nine decision matrices was constructed as follows: four of the eight rows are consistent with four major decision theoretic criteria -- maximax, maximin, expected value (in the Laplace sense), and minimax regret (see Luce and Raiffa, 1957). For example, in Table 1, row S_1, since it has a payoff of 12 (larger than all other payoffs in that table), corresponds to the maximax strategy. Row S_4 corresponds to the maximin strategy, row S_8 to the expected value strategy (again, assuming a uniform probability distribution), and row S_5 to the minimax regret strategy. The other four rows correspond to suboptimal choices: each of these rows is dominated by at least one of the four optimal rows. In Table 1, row S_1 dominates row S_3 ($12 > 11$, $4 = 4$, $4 > 3$, and $0 = 0$), S_4 dominates S_6 ($6 > 5$, $6 > 5$, $4 = 4$, and $4 = 4$), S_8 dominates S_2, and S_5 dominates S_7.

The order of appearance of the eight types of rows in each of the nine matrices was randomized as was the order of payoffs within each row. (An exception is the minimax regret row and its associated dominated row, since the regret calculation depends upon other payoffs that appear in the same column.) The order of presentation to S of the nine decision matrices was sequential: 1, 2, ..., 9. Matrices 1, 4, and 7 had all positive payoffs; matrices 2, 5, and 8 had both positive and negative payoffs; and matrices 3, 6, and 9 had all negative payoffs. Additionally, matrices 7, 8, and 9 had significantly larger numeric entries than matrices 1 through 6 (an average of 9.23 versus 5.29, respectively).

The subject could exhibit inconsistent choice behavior using these matrices (he could select a row corresponding to one decision criterion on one matrix and corresponding to a different decision criterion on another matrix) and/or suboptimal choice behavior (he could select a row dominated by another). Previous experimentation (Tuggle, 1972) shows that subjects (undergraduate and graduate students and practicing managers) exhibit both of these behaviors on these very problems. However, this S, in fact, makes no suboptimal choices and evidences only one inconsistency (see Section III). Yet, as Sections III, IV, and V will detail, his choice processes differ substantially from those dictated by decision theory. (See Luce and Raiffa, 1957, for a statement of normative decision theory.)

B. The Subject

The only subject studied in this paper was a male, first-year M.B.A. candidate at the University of Kansas who had not had courses in operations research or in decision theory. He was invited directly by one of the authors (FHB) to participate in an experiment on decision-making, and S was promised that he could either receive a flat payment of $2 for participation or gamble based on the decisions he would make. We informed S, if he chose to gamble, that after he had made his nine decisions, we would apply some unspecified generating processes to select one of the matrices and to select a state of nature. He did not have to announce whether he wanted to gamble or not until he had seen all nine tables and made his choices. (The subject did decide to gamble and won an additional $16.)

III. Protocol Analysis

Limits to the space available here prohibit us from presenting and analyzing S's entire protocol and Problem Behavior Graph (PBG), a time-ordered graph of S's verbal behaviors that are then to be simulated by the IPS (Newell and Simon, 1972). The complete protocol and PBG are available from the authors. Excerpts from S's protocol and his PBG are presented in Figure 1, primarily on payoff matrix 1 as illustrated in Table 1.

A. Crude Generalizations

Roughly, the behavior of S over all nine matrices seemed to be as follows: He first labeled the table as to whether it is all positive, mixed, or all negative (presumably doing a quick scan of the payoffs), although his later actions are not differentiable based on the label given.

Second, he sequentially searched the action alternatives open to him, fixating on those that have a distinguishing characteristic (e.g., one very large or very small payoff, or a number of "strong" or "weak" payoffs).

Third, he partitioned, explicitly and implicitly, his eight action alternatives into three sets (while sequentially examining them): those that he dislikes (a "reject" list), those that appeal to him (a "consider" list), and those that received no verbal indications (an "ignore" list).

Fourth, a choice was made from those alternatives that are present on the "consider" list. In the case of numerous alternatives, a pairwise comparison and rejecting process is used. From S's protocol, the data led us to infer the following comparison process:

(1) S never talked about computing a sum (row sum or column sum) or an expected value.

(2) Of all the numbers S verbalized, he never verbalized one that was close to a sum or

an expected value.

(3) S did verbalize on several occasions about comparing "... these possibilities across the board" (emphasis added).

From these data, we infer that S was doing some sort of column-by-column comparison of the two rows in question. Simply to have some definite procedure to follow, we devised a process that compares corresponding payoffs in each of two rows and that rejects the row that has fewer dominating payoffs. (This process is explicated later in Section IV, Part C, where SIDIP's CHOICE subroutine is discussed.)

Finally, it is instructive to mention the other types of verbal behavior present in S's protocol. Besides the anticipated vague statements, expressed confusions, and overt catharses, S did engage in a singular behavioral pattern, from time to time. Particularly on matrix 7 and to a lesser extent on matrices 1, 2, and 8, S spent some time (2, 5, 10, and 6 protocol lines on matrices 1, 2, 7, and 8, respectively) attempting to isolate identifying characteristics of the four columns so as to be able to assign subjective probability estimates to them. For example, on matrix 7, S noted that in one column there is a larger proportion of the larger payoffs in the matrix, and he tentatively concluded that that column had a lesser chance of occurring. However, in each case, S apparently eventually discarded that "column-processing" line of analysis and continued his original "row-processing" type of analysis.

B. Examination of PBG

Problem Behavior Graphs (PBGs) are concise ways of encoding the dynamics in the problem-solving and decision-making behaviors present in a protocol. (See Newell, 1966, for a complete discussion of the construction and utility of PBGs.) The nodes in such a graph are the actions or statements verbalized by the subject, either presented verbatim or paraphrased. The nodes are interconnected by lines (arcs or edges of the graph), which put a time ordering on the graph: time runs (first) to the right and (then) downwards. (The reason for allowing time to run to the right is to allow succinct presentation of episodic exploration on the part of the subject.)

Figure 1 presents the PBG over decision situation 1 that we developed and used in our study of S. In this PBG, there are behaviors consistent with our generalizations of Part A, behaviors inconsistent with those generalizations (but not necessarily inconsistent with our detailed model in the next Section), and behaviors not included in those generalizations.

The consistent behaviors are (1) the matrix is correctly labeled as being an all-positive (or, more accurately, all-nonnegative); (2) only those rows with double-digit entries (S_1, S_3, and S_5) are "considered;" the rest are ignored;

All positive here on the first table
|
\$12, \$0, \$10, \$4, and \$8
|
Note 12 in S_1 -- Consider
|
Note 10 in S_5 -- Consider
|
Note ? in S_3 -- Consider
|
Note 0 in S_1 -- Note 0 in S_3
|
XEQ column looks larger
|
S_1 -- more money -- but has a 0 -- Discard
|
S_3 -- also has a large outcome -- but a 0 -- Discard
|
S_5 -- has a large outcome
|
Accept S_5

Figure 1: Problem Behavior Graph Over Decision Situation 1

(3) S_1 and S_3 are "rejected" for having the table minimum, namely, a payoff of zero.

The inconsistent behaviors are (1) S_5 is "considered" before S_3, violating our sequential row-processing hypothesis; (2) column processing may be going on when S verbalizes "\$12, \$0, \$10, \$4, and \$8", as all of these payoffs are to be found in column ZEJ (see Table 1). Alternatively, S could be reading parts of S_1, S_5, and S_8, doing a row-processing analysis, or doing something altogether different.

A behavior by S not present in our generalization is S's observation that the "XEQ column looks larger." In fact, it is larger than the WUH and QUG columns, but only equal to the ZEJ column. Our subject apparently did not notice or utilize this bit of information.

Examination of all nine PBGs yields some evidence -- pro, con, and irrelevant -- to the preceding crude generalizations. But this is not the point. The question is how well our detailed computer simulation model of Section IV matches the choices and significant processes of the nine PBGs. This question is answered rhetorically in Section V.

IV. Simulation Program -- SIDIP

Our computer simulation program (SIDIP, an acronym for Simulation of Individual Decisions through Information Processing) is written in L6 (see Knowlton, 1966, for an explanation of the Bell Telephone Laboratory's Low Level Linked-List Language) for the Honeywell 635 computer. The program and job cards occupy approximately 500 card images, and the data take 40 card images. (Complete listings of both are available from the authors.)

A. Data Structures

The primary reason behind the selection of L6 as our language was its ability to construct and manipulate complex data structures. According to Newell and Simon (1972, pp. 19ff.), there are three important aspects of a simulation program of human thought: a set of symbol structures, encoding the information present in S's short-term, long-term, and external memories (covered in the remainder of this part of the paper); a set of Elementary Information Processes (EIPs) with which to operate on the symbolic information (presented in Part B); and a program, an ordered collection of EIPs, organized to accomplish a whole task (presented in Part C).

The major symbol structure posited for our S is an encoding of the payoff matrix, replete with row, column, and table description lists. The matrix is represented as a 32 node network (8 rows by 4 columns). Nodes in the same row are doubly-linked (from a payoff in the matrix, one can access its right neighbor or its left neighbor); likewise, nodes in the same column are doubly-linked. Besides the node linkages, the following node information is loaded and is static throughout a run: its row number, its column number, its value and its sign.

The following descriptive information is dynamically created during the execution of SIDIP: the matrix is labeled as all positive, all negative, or mixed; maximum and minimum payoff values for the matrix are computed. The number of negative values in the table is computed. Nodes are labeled as double-digit if their value is larger than 9. Both rows and columns have these attributes created: maximum value, minimum value, count of the number of negative numbers, and count of the number of double digit numbers. Finally, each row is identified by its status or evaluation during the decision process: "Reject," "Consider," "Good," or "Accept."

B. EIPs

In one sense the Elementary Information Processes we presume are the legal L6 commands, and in another sense they are the seven types of EIPs specified by Newell and Simon (1972, pp. 29-30): Discrimination, Tests and Comparisons, Symbol Creation, Writing Symbol Structures, Reading and Writing Externally, Designating Symbol Structures, and Storing Symbol Structures. More specifically, SIDIP is based upon (and implicitly, we presume S has the capabilities of) these EIPs: the ability to create attribute information such as described in Part A, the ability to retrieve, compare, and distinguish such information, the ability to input (hear or read, for S and SIDIP, respectively) and to output (speak or print) symbolic information, the ability to do simple numeric processing (e.g., to add two numbers, to recognize that $7 > 5$ and that $-4 < -3$, to save intermediate results), and the ability to interpret a program of EIPs. These are the only EIPs we require of SIDIP, and the only ones we posit about S. The sufficiency of these EIPs for SIDIP will be empirically demonstrated indirectly and implicitly in Section V. The necessity of these EIPs for S does not violate anything we now know about the cognitive capabilities of humans.

C. Macroprograms

In this part we shall indicate how SIDIP operates, primarily by reference to its flowcharts in Figures 2, 3, 4, and 5 and by reference to the output it produces, such as in Figure 6. SIDIP can be conceptualized as a main program (Figure 2) and three subroutines: EVALUATE (Figure 3), CHOICE (Figure 4), and COMPARE (Figure 5). The main program initializes data areas, reads and echo prints the decision matrices, does some background information processing, then calls on EVALUATE to sequentially examine and label each row (as "Considered," "Rejected," or ignored), and finally calls on CHOICE to determine the row to be "Accepted."

The EVALUATE subroutine examines each row in sequence for the properties listed in Figure 3 and makes an evaluation based on the presence or absence of those properties. Like our human subject, EVALUATE "states" (prints) what its evaluation of a row is (as soon as one is made) and also "states" (prints) what the reasons for the evaluation are. (If both a "consider" and a "reject" evaluation is made about a row, only the one made earlier is retained.)

Two of the terms in Figure 3 are vague: "strong entry" and "weak entry." Our operational definition of these terms, based upon S's verbalizations, follow. A "strong entry" is a double-digit payoff, when the matrix is not all negative and when the proportion of double-digit payoffs in the table is 25% or less. In all other cases, a payoff is "strong" if it is one of the top three payoffs just below the maximum of the matrix.

A "weak entry" is a negative payoff, in the case of a mixed matrix. Otherwise, a payoff is "weak" if it is one of the two payoffs just immediately above the minimum of the matrix. The CHOICE subroutine simply reorders the payoffs in each row and continues to call COMPARE until only one row remains "considered." The remaining row is "accepted."

The COMPARE subroutine performs a pairwise comparison of two (internally ordered) rows. A count is kept of the number of times one row's entries dominate the other row's. The row whose total count is smaller (if either is) is labeled "reject" and control returns back to the CHOICE subroutine.

(One may inquire what action SIDIP takes when two rows are labeled "consider" and neither one dominates the other. In this case, SIDIP would apparently be in an infinite loop; noting such cases, SIDIP "accepts" all such rows. Our

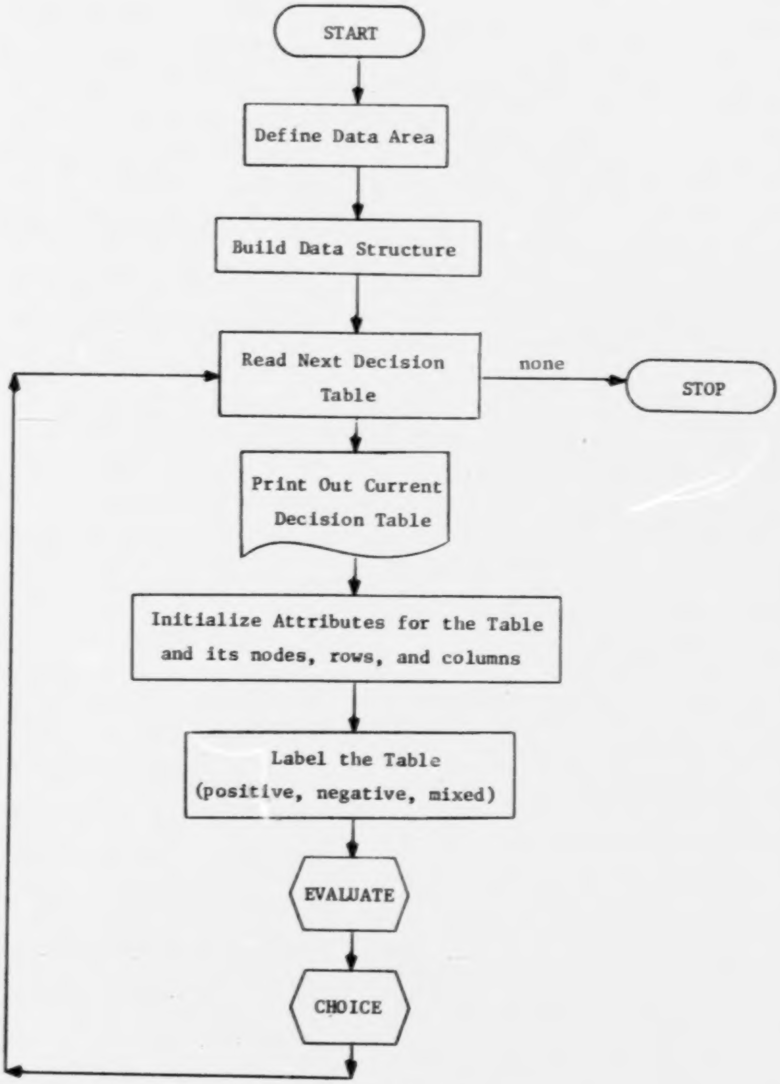

START
Define Data Area
Build Data Structure
Read Next Decision Table
none
STOP
Print Out Current Decision Table
Initialize Attributes for the Table and its nodes, rows, and columns
Label the Table (positive, negative, mixed)
EVALUATE
CHOICE

Figure 2: Main Flowchart of SIDIP

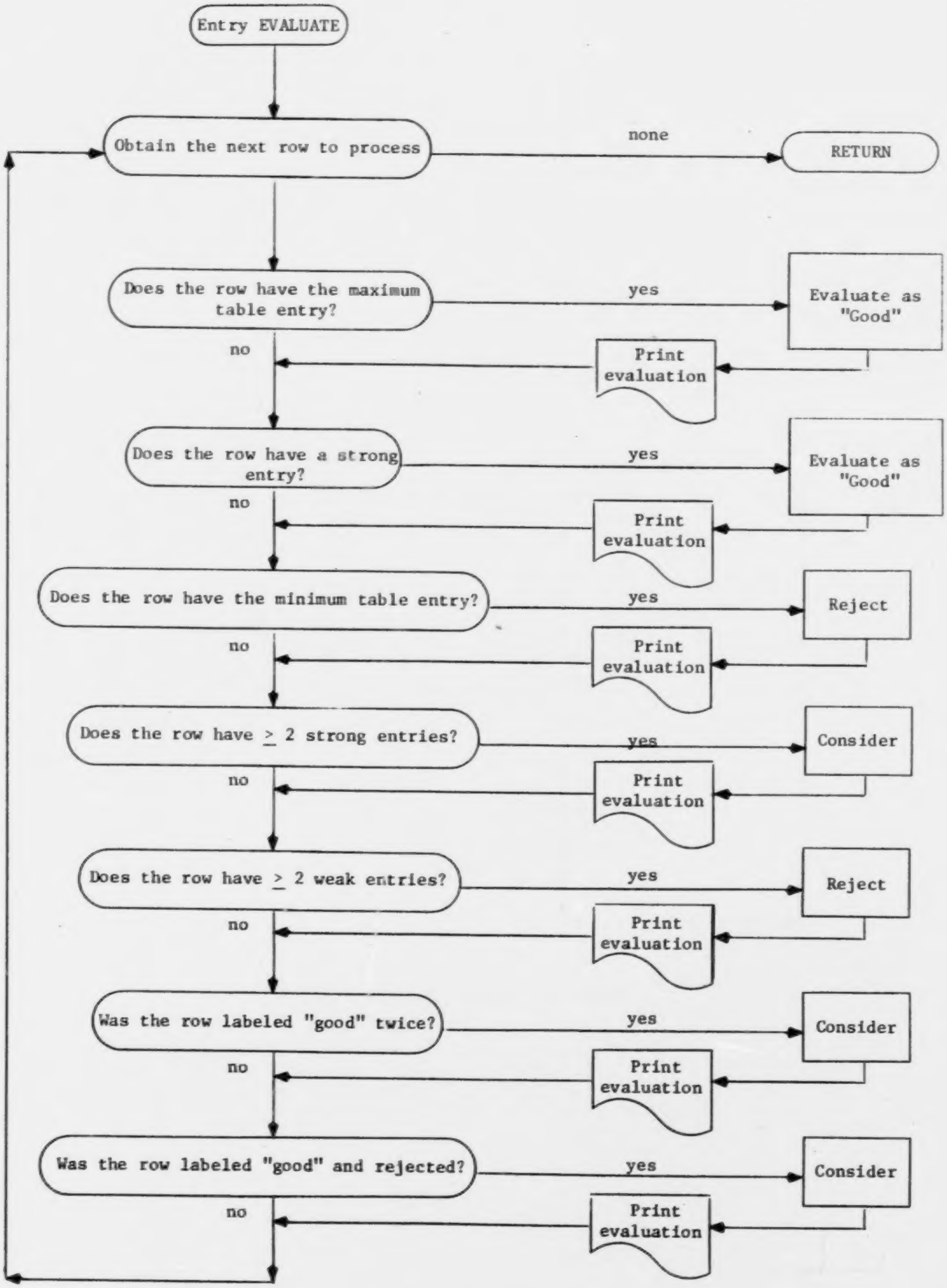

Entry EVALUATE
Obtain the next row to process
none
RETURN
Does the row have the maximum table entry?
yes
Evaluate as "Good"
no
Print evaluation
Does the row have a strong entry?
yes
Evaluate as "Good"
no
Print evaluation
Does the row have the minimum table entry?
yes
Reject
no
Print evaluation
Does the row have ≥ 2 strong entries?
yes
Consider
no
Print evaluation
Does the row have ≥ 2 weak entries?
yes
Reject
no
Print evaluation
Was the row labeled "good" twice?
yes
Consider
no
Print evaluation
Was the row labeled "good" and rejected?
yes
Consider
no
Print evaluation

Figure 3: SIDIP Subroutine EVALUATE

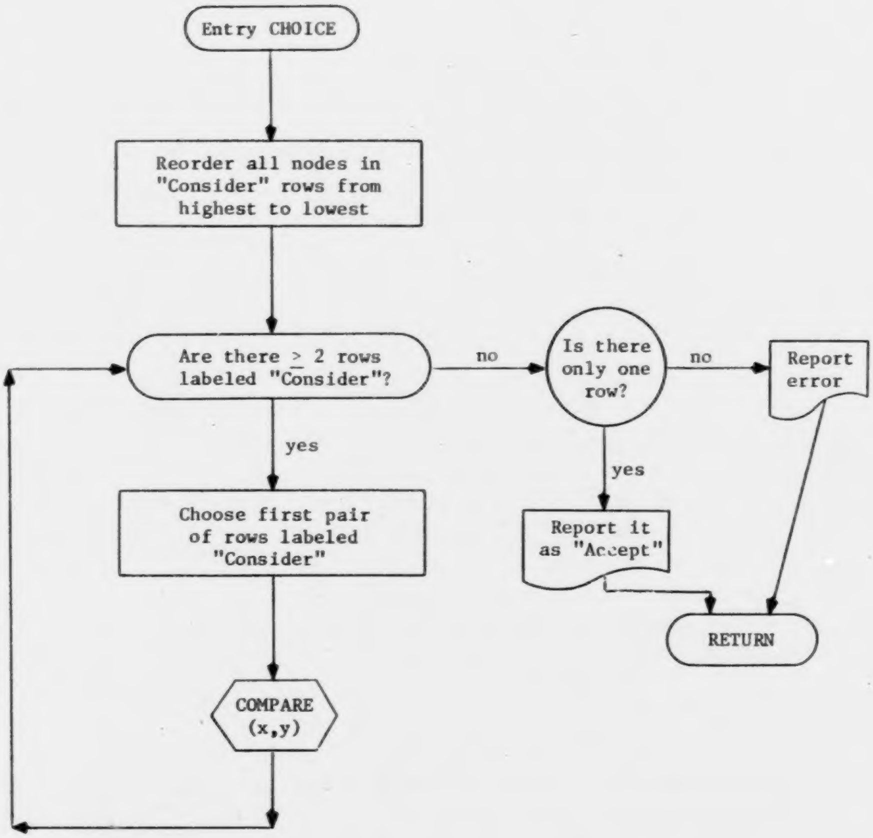

Entry CHOICE
Reorder all nodes in "Consider" rows from highest to lowest
Are there ≥ 2 rows labeled "Consider"?
no
Is there only one row?
no
Report error
yes
yes
Choose first pair of rows labeled "Consider"
Report it as "Accept"
RETURN
COMPARE (x,y)

Figure 4: SIDIP Subroutine CHOICE

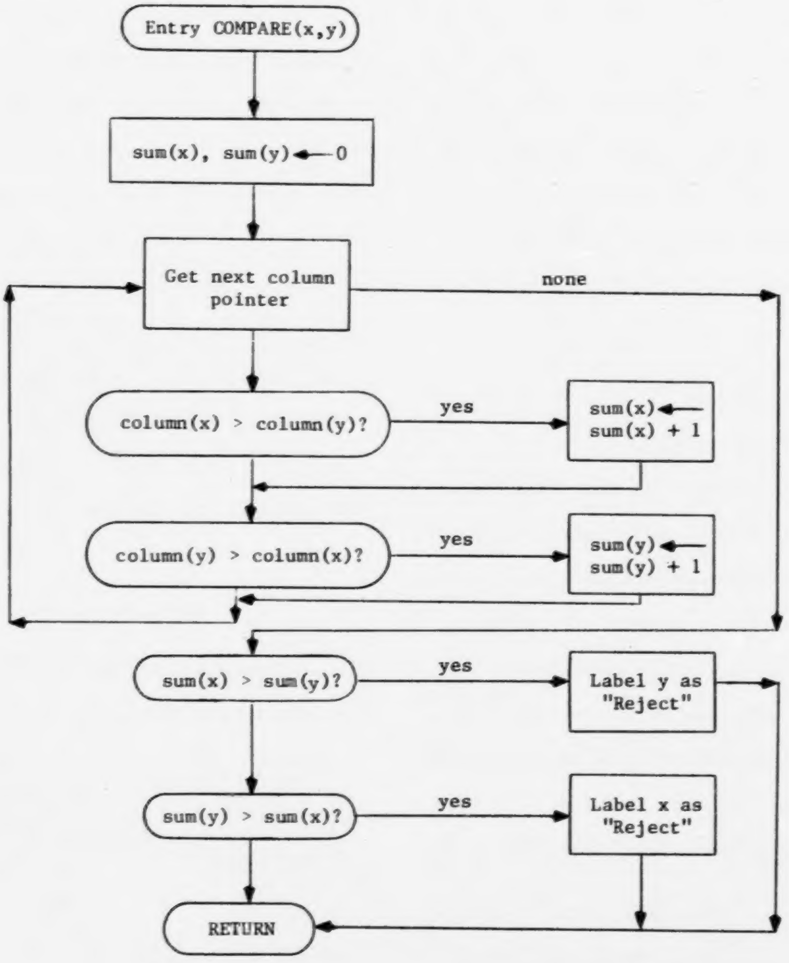

Entry COMPARE(x,y)
sum(x), sum(y) ← 0
Get next column pointer
none
column(x) > column(y)?
yes
sum(x) ← sum(x) + 1
column(y) > column(x)?
yes
sum(y) ← sum(y) + 1
sum(x) > sum(y)?
yes
Label y as "Reject"
sum(y) > sum(x)?
yes
Label x as "Reject"
RETURN

Figure 5: SIDIP Subroutine COMPARE

human subject never mentioned facing such a situation, but SIDIP encountered it in four cases.)

Finally, in Figure 6, one can see what is printed out by the computer as it processed decision situation 1: an echo print of the matrix, a labeling of the matrix, an evaluation of the rows, and a final choice. These latter three elements effectively constitute a trace or a protocol of SIDIP's behavior, which will be compared with S's protocol (PBG, really) in the next section.

V. Results and Analysis

A. Performance of SIDIP

When confronted with the same nine decision matrices as the subject, SIDIP made exactly the same choices on four matrices, made an incorrect choice on one matrix, and could not choose between two rows on each of four matrices -- but in these four cases, one of the two undecided rows was always the one selected by the subject. In a sense, then, SIDIP was "right" four times, "wrong" once, and "half-right" four times. The significance of these findings is examined in Part B. A report of the test of the decision process appears in Part C.

It may interest the reader to know that SIDIP's wrong choice occurred on decision situation 5 (mixed payoffs) where S chose the expected value row[*] and SIDIP chose the maximin row. In this case S explicitly did not reject the row having the minimum table value. The "half-right" choices occurred on matrices 3 (all negative), 7 (all positive), 8 (mixed), and 9 (all negative). S chose the expected value row in these four cases; SIDIP also chose the expected value row and respectively chose the regret row, the maximin row, the maximax row, and the maximin row in addition.

B. Performance Tests

To gain insight into the ability of SIDIP to simulate S's decisions, we shall compare its

This table is all positive

|

Row S_1 looks good because of the maximum table value

|

Row S_1 looks pretty good, large numbers

|

Row S_3 looks pretty good, large numbers

|

Row S_5 looks pretty good, large numbers

|

Row S_1 looks bad since minimum table value

|

Row S_3 looks bad since minimum table value

|

My choice is S_5

Figure 6: PBG of SIDIP Over Matrix 1

---

[*] We shall continue to confuse the maximum expected value criterion and the Laplacian criterion -- which presumes a uniform probability distribution.

choices to those of two models with random selection mechanisms.

The first model presumes choices are made by a random selection from the eight rows on a matrix with a uniform probability of $p = 1/8$. Since there are nine independent trials (decision situations), the Bernoulli assumptions apply, and we can derive two essential characteristics about our model:

$n = 9$, $p = 1/8$, and $q = 1 - p = 7/8$

therefore, the mean number of correct choices would be

$\mu = np = 9 \cdot 1/8 = 1\ 1/8$

with a standard deviation of

$\sigma = \sqrt{npq} = \sqrt{9 \cdot 1/8 \cdot 7/8} \cong 1.$

But SIDIP makes either 6 correct choices (4 fully correct plus 4 half-correct) or 4 correct choices, depending on how one chooses to treat the four "half-right" choices. This yields a Z score ($Z = \frac{x_1 - x_2}{\sigma}$) of either

$$Z_6 = \frac{6 - 1\ 1/8}{1} = 4.875 \quad \text{or}$$

$$Z_4 = \frac{4 - 1\ 1/8}{1} = 2.875$$

The interpretations are that the Z_6 score is 4.875 standard deviations better than the random model (statistically significant at $\underline{p} < .001$), and the Z_4 score is 2.875 standard deviations better (significant at $\underline{p} < .003$). A similar analysis assuming random choice only occurs among the four nondominated alternatives yields

$n = 9$, $p' = 1/4$, $q' = 1 - p' = 3/4$

with resulting changes in $\mu = np' = 9 \cdot 1/4 = 2\ 1/4$ and $\sigma' = \sqrt{np'q'} = \sqrt{9 \cdot 1/4 \cdot 3/4} \cong 1.3$ with corresponding Z scores of

$$Z'_6 = \frac{6 - 2\ 1/4}{1.3} \cong 2.88 \quad \text{and}$$

$$Z'_4 = \frac{4 - 2\ 1/4}{1.3} \cong 1.35.$$

Z'_6 is significant at $\underline{p} < .003$, and Z'_4 is significant at $\underline{p} < .1$.

Since we feel that it is too harsh to assume that a "half-right" choice is entirely wrong, we are forced to judge SIDIP's choices based on the Z_6 and Z'_6 scores. Consequently, we conclude that SIDIP's performance is significantly superior to the choices generated by these two random models of decision-making.

Next we could test "as if" choice behavior. The null hypotheses become S chooses "as if" he is using a maximax (or maximin, or minimax regret, or Laplace -- expected value) criterion. We cannot reject such hypotheses on a purely statistical basis (unless something like a strong inference point of view is accepted as in Barron, 1970) since we have no appropriate error theory and thus, no acceptable statistical methodology.

Rather than argue on a statistical basis we would suggest that S's protocols clearly show that since none of these decision models (random, maximax, etc.) is determining S's choices a more complex model such as SIDIP is required.

C. Test of Process Similarity

Turing's test (Turing, 1963) is a classical technique in the field of artificial intelligence by which one tests the processes in a program that allegedly simulates a part of human cognition. For reasons best illustrated by reference to Table 2, Turing's test is not completely applicable in this situation: S's protocol omits inarticulated processes. Thus, S's PBG is necessarily incomplete. In addition, S's PBG includes unnecessary or irrelevant behaviors as well as inconsistencies (errors). SIDIP is necessarily consistent, thus introducing perhaps insignificant processes (from a decision theoretic point of view). To test for congruence in the structures of the two PBG's, we quite arbitrarily aggregated all row evaluations and comparisons (calling them the "total behaviors" in Table 2), ignored all other behaviors, and then compared all of the "total behaviors" of SIDIP to the "total behaviors" of S. Since presumably S neglected to mention some of his thoughts, his "total behavior" is necessarily less than that of SIDIP, which we can force to be completely verbal. This does mean, though, that for many of SIDIP's behaviors, there will be neither collaborative nor disconfirming evidence present in S's (skimpy) PBG. Of the 50 row evaluations and comparisons by SIDIP which can be tested by behaviors of S, 39 of them (78%) agree with behaviors by S and 11 disagree. There are more behaviors by SIDIP that agree with behaviors of S than disagree in each of those nine cases, so SIDIP seems to be uniformly good. To get some feeling for the significance of the 78% correct figure, examine the following naive random model: suppose that this model either emits a correct behavior or an incorrect behavior with a uniform probability of 1/2. This is very conservative, because there are so many ways a row evaluation can be wrong (SIDIP's evaluation of a row can agree or disagree with S's evaluation; even if

Matrix	Total SIDIP Behaviors	Agreement	Disagreement	Absent	Total S Behaviors
1	9	8	0	1	8
2	19	7	3	9	10
3	9	2	1	6	5
4	14	3	1	10	4
5	9	4	2	3	9
6	10	1	0	9	4
7	24	8	3	13	12
8	11	3	1	7	4
9	11	3	0	8	5
Total	116	39	11	66	61
% of column 1		33.6%	9.5%	56.9%	52.6%

Table 2: SIDIP's PBG Compared to S's PBG

there is agreement, SIDIP is wrong if its reasoning differs from S's) and because there are three -- not two -- possible results from a comparison (a preferred to b, b preferred to a, indifference between a and b). Nevertheless, with that assumption and by assuming that the 50 behaviors are independent, we can again apply the Bernoulli model to derive the mean number correct (from the random model). This mean is $\mu = np = 50 \cdot 1/2 = 25$ with a standard deviation of

$$\sigma = \sqrt{npq} = \sqrt{50 \cdot 1/2 \cdot 1/2} = \frac{\sqrt{50}}{2} \cong 7/2 = 3\ 1/2.$$

The score for the behaviors of SIDIP is $Z = \frac{39 - 25}{3\ 1/2} = \frac{14}{3\ 1/2} = 4$, which is statistically significant at $p < .001$. Thus, SIDIP's intermediary behaviors were significantly closer to S's behaviors than this simple random model.

VI. Discussion

A. Conclusions

According to Van Horn (1971), there are several ways by which one can validate a computer simulation experiment: use models with high face validity, run "Turing" type tests, etc. (Van Horn lists several other techniques). The statistical tests that have been run so far on SIDIP suggest our simulation is significantly better than random decision models. The numbers and kinds of articulated behaviors summarized in Table 2 suggest that simple decision models such as maximax, Laplace, etc., are inadequate. As indicated SIDIP's intermediary behaviors are significantly closer to S's behaviors.

Mihran (1972) presents several procedures for verifying and validating both deterministic and stochastic computer simulation programs. However, none of these tests are really appropriate for protocol simulations. For this reason we have not further tested the structural congruence of the PBGs of SIDIP and our S.

There are some deficiencies in SIDIP. SIDIP makes one wholly incorrect choice, and four others are only partially correct. Eleven of SIDIP's 50 applicable behaviors are wrong, and 22 of S's 61 decision-making behaviors (36%) remain unexplained by this version of SIDIP. Thus we conclude that SIDIP explains reasonably well the nucleus of our subject's decision-making processes, but that there are still peripheral processes of importance by S that the current SIDIP does not capture.

B. Incremental Improvement in Decision-Making

Given the caveats in the previous part, it is obviously premature to press forward strongly in the area of improving decision-making processes by studying simulations of individuals. In order to conclude the research thrust begun in this paper, though, we shall pretend that SIDIP is near 100% successful to sketch the remaining work to be accomplished.

Assuming (heroically) that SIDIP adequately simulates the decision processes of S, we can now perform experiments upon the computer simulation program. Suppose, by way of illustration, that S articulates a desire to behave in a manner consistent with the maximize expected value criterion, but S does not consistently do so. Then an easy way to change SIDIP so that it behaves in that manner is to alter the COMPARE subroutine: after reordering the columns within the two rows (from highest payoff to lowest), do not compare the columns by simply noting "above," "below," or "equal." Instead, determine and record how much above or below one row's column is over the other. Total these differences, and the row with the higher sum is then the row with larger expected value.

Once it is learned that the suggested changes in COMPARE cause SIDIP to behave in the desired manner, then this information can be presented to S. By allowing S to learn of his shortcomings or through some similar procedure, improvements in S's decision-making behavior may be incrementally introduced. (The design of an acceptable training procedure is still an unsettled issue.) He retains the familiar and comfortable essentials of his decision-making process, but his decision-making ability is now improved.

C. Future Work

There are basically four avenues along which this research should be continued: First, improvements in SIDIP need to be made so that it simulates S's behaviors even more closely. For example, S clearly expends much effort in attempting to differentiate columns. SIDIP should also look for regularities or peculiarities in columns and then introduce corresponding changes into the subjective probabilities associated with those columns.

Second, protocols from more subjects should be collected so that more can be learned about the actual decision-making processes of humans.

Third, experiments should be conducted to learn a training procedure that is successful at modifying decision-making processes.

Fourth, this entire paradigm should eventually be moved out of the laboratory and into the real world. Ultimately, it is not the decision-making processes of college students that one is interested in studying, simulating, and improving, but rather the decision processes of military leaders, government officials, and business executives.

Bibliography

Barron, F. H., "The Potential Adopter's Decision Rule: A Constrained Optimization Model of Decision Making Under Risk and Uncertainty," Unpublished doctoral dissertation, University of Pennsylvania, 1970, and Working Paper No. 35, Department of Management Sciences, University of Waterloo, June, 1970.

Knowlton, K. C., "A Programmer's View of L6," Communications of the ACM, Vol. 9, No. 8, Aug., 1966, pp. 616-625.

Luce, R. D. and Raiffa, H., Games and Decisions, Wiley, New York, 1957.

MacCrimmon, K. R., "Descriptive and Normative Implications of the Decision Theory Postulates," in Borch, K. and Mossin, J. (eds.), Risk and Uncertainty, St. Martin's Press, New York, 1969.

Mihram, G. A., "Some Practical Aspects of the Verification and Validation of Simulation Models," Operational Research Quarterly, Vol. 23, No. 1, Feb.-March, 1972, pp. 17-29.

Newell, A., "On the Analysis of Human Problem-Solving Protocols," paper delivered at the International Symposium on Mathematical and Computational Methods in the Social Sciences, Rome, Italy, July 4-9, 1966.

Newell, A. and Simon, H. A., Human Problem Solving, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

Tuggle, F. D., "Decision Making Under Uncertainty: Empirical Tests of Normative Theory," Journal of Business Administration, in press, 1972.

Turing, A. M., "Computing Machinery and Intelligence," in Feigenbaum, E. A. and Feldman, J. (eds.), Computers and Thought, McGraw-Hill, New York, 1963.

Van Horn, R. L., "Validation of Simulation Results," Management Science, Vol. 17, No. 5, Jan., 1971, pp. 247-258.

Identification of Viable Biological Strategies for Pest Management by Simulation Studies

W. W. Menke
Associate Professor of Management
College of Business Administration
Matherly 325
University of Florida
Gainesville, Florida 32601

Abstract

Interdisciplinary research by management scientists and entomologists at the University of Florida has developed a stochastic computer model for studying interactions between an insect population, its host food crop, and other variables. This population growth model, highly adaptable to any insect and any host crop, is technically characterized by discrete arrivals, infinite servers and multi-stage, continuous service-time distribution functions.

Because steady state is seldom achieved in nature, this paper identifies combinations of critical starting conditions (number of insects, disparate start times for insects and host crops) and critical stages for induced survival rate reductions to minimize crop damage. Sensitivity analyses serve to identify the most promising areas for future entomological research in pest management strategies.

I. Introduction

This research was initiated in the belief that management science could and should contribute to the solution of ecological problems. One of the most serious of these is the ecological consequences of the unrestricted use of pesticides. Pest management addresses the problem by identifying and testing alternate strategies which would minimize the use of pesticides for crop protection while maintaining the crop's economic worth.

The experiments reported here were performed in response to the following general questions.

> If one has a simulation model of the damage caused by a specific pest on a specific valuable food crop, is there a stage in the pest's life cycle where a fixed decrease in pest population is most effective in minimizing crop damage?
>
> Would the implications of the results of the simulations identify useful strategies for minimizing the application of ecologically dangerous pesticidal materials?

II. The Ecosystem Being Simulated

The ecosystem simulated for the experiments of this report is that of a soybean host crop infested by the velvet bean caterpillar (VBC), one of the major pests for soybeans in North Florida. Soybeans were selected as a host crop because of their worldwide economic importance as a high protein food. [1]

The VBC moth overwinters in South Florida and is suspected of invading Florida from the Caribbean area each year. The adult female moth lays its eggs and about two generations of insects develop during the year until death in late fall (or out-migration of the adult moth) reduces the population to essentially zero in North Florida soybean fields.

The eggs, 800 on the average per female, are laid individually by the adult over a period of about eight days. After hatching, the insect passes through six stages of growth (instars) as caterpillars and a pupae stage before emerging as an adult moth to mate, disperse, and lay eggs for the next generation. The length of time the caterpillar remains in an instar (dwell time) is a stochastic variable influenced by the environmental conditions during the stage. Available field experimental data indicate the dwell time probability distribution to be normal for all instars. Each, however, has a different average and standard deviation. The VBC causes damage by defoliation of the soybean plant, the amount varying with the instar.

The soybean plant was considered to have ten stages of development from emergence to unifoliate leaves through maturity. The plant's leaf development between the critical stages in plant maturity, podset to podfill, and the per cent defoliation by the pest during this period was of primary interest in the model since economic crop damage occurs when threshold defoliation levels are exceeded here.

III. The Ecosystem Model

The simulation model of the insect population dynamics, the growth of the soybean crop, and the damage by defoliation caused by the VBC was developed through the interdisciplinary efforts of a management scientist and three entomologists. Initially knowledge about each others' discipline was minimal. General concepts of model structure emerged during repeated conferences which were mutually educational in biology, agronomy, entomology, management science, and modeling techniques. Specific concepts used in the model structure are described below.

Technically, the insect population dynamics is modelled in terms of stochastic variates characterized by discrete arrival times (the adult moth invasion), a discrete starting population (the eggs), infinite servers (the life cycle for each insect) and multi-stage service time (the continuous distributions for dwell time at each instar.)

The crop development portion of the

model is a deterministic function of time using EV [leaf area] since the leaf area per acre is so large (even at the seedling stage, leaf area is approximately 1×10^6 cm^2/acre). Critical points for the functional relation are seedling, midbloom, podset and podfill. All soybean leaf area growth and defoliation by the VBC are expressed in cm^2 per acre. Insect population counts and leaf damage are calculated at weekly intervals after the date of simular planting.

The model is presently one-dimensional in that it does not consider in-migration or out-migration of the VBC after the starting moth invasion. Nor does it consider the spread or diffusion of the insect population within the field from the point of contact of the initial invading moths. Therefore, simulated population counts and leaf defoliation calculations are average values per acre, ignoring localized hot spots which are expected in the real situation.

FORTRAN IV was chosen as the computer language for the simulation in the belief that it would be more flexible in permitting future expansion of the model to include the above considerations. In addition FORTRAN flexibility would be useful when non-ideal environmental effects (temperature, humidity, day length, rainfall, surround, etc.) were included. It is anticipated that SIMSCRIPT would be used at a more refined stage in the model development.

Biological and entomological data needs emerged as the model was developed. These data were collected in a variety of ways. In some cases, e.g., dwell time per instar, literature search combined with recent experimental results provided the answer; in some cases, e.g., average leaf area eaten per instar per caterpillar, private communications with researchers yielded unpublished results; in other cases, e.g., typical moth invasion dates and crop development data, PERT type "min. - expected - max." questions [2] were posed to the entomologists; in a few cases, e.g., survival rate per instar, little hard experimental information was available and reliance was placed upon general knowledge and intuition for reasonable values.

IV. Operation of the Model

Figures 1 and 2 show simplified flow chart descriptions of the operation of the model.

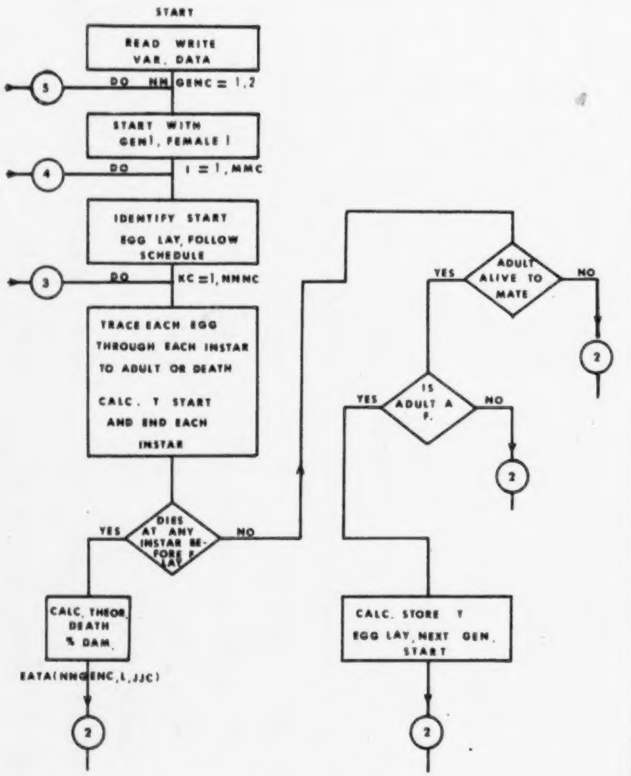

START
READ WRITE VAR. DATA
DO NNGENC = 1,2
5
START WITH GEN1, FEMALE 1
DO I = 1, MMC
4
IDENTIFY START EGG LAY, FOLLOW SCHEDULE
DO KC = 1, NNNC
3
TRACE EACH EGG THROUGH EACH INSTAR TO ADULT OR DEATH
CALC. T START AND END EACH INSTAR
DIES AT ANY INSTAR BEFORE F LAY
YES
NO
CALC. THEOR. DEATH % DAM.
EATA(NNGENC,L,JJC)
2
ADULT ALIVE TO MATE
YES
NO
2
IS ADULT A F.
YES
NO
2
CALC. STORE T EGG LAY, NEXT GEN. START
2

Figure 1

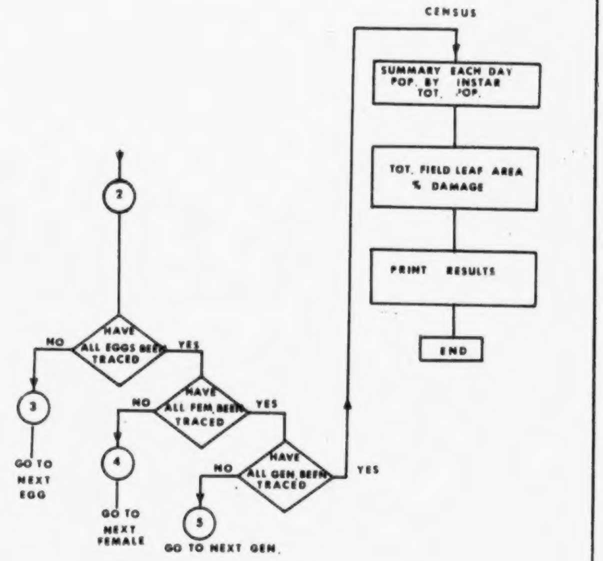

2
HAVE ALL EGGS BEEN TRACED
NO
YES
3
GO TO NEXT EGG
HAVE ALL FEM BEEN TRACED
NO
YES
4
GO TO NEXT FEMALE
HAVE ALL GEN BEEN TRACED
NO
YES
5
GO TO NEXT GEN.
CENSUS
SUMMARY EACH DAY POP. BY INSTAR TOT. POP.
TOT. FIELD LEAF AREA % DAMAGE
PRINT RESULTS
END

Figure 2

The model traces the development of each egg from each female adult through the instar in which death occurs or until the larva becomes an adult moth. If the adult is a female, the time of egg laying for the next generation of progeny is calculated and recorded. After all eggs from all females of one generation are traced, the program automatically repeats the calculations for the next generation of caterpillars until the required number of generations have been traced.

Monte Carlo techniques are used to calculate survival at each stage of development from egg through adult moth, the time duration a particular individual remains in a particular instar (dwell time), as well as whether the adult is a male or female. Thus 20 different random number series were identified and used for each simulation run. Subsequent simulation runs used different random number seeds, themselves chosen from random number tables for each of the 20 random number series of the run.

After all eggs from all females for all generations have been traced, the program prints out a census calculated at weekly intervals from the starting date for egg laying for the first generation. The census details for each day of count:

1. the population in each instar,
2. the total population including eggs,
3. the total population of larvae only,
4. the cumulated soybean leaf area eaten by all the caterpillars through this day,
5. the crop leaf area available,
6. the per cent of the available leaf area eaten.

Table 1 describes the variables of the model. These are classified according to type (dependent or independent), description, source for numerical values, and if data, the basis for the data.

Figure 3 indicates the assumed dwell-time distributions, the egg laying schedule and the eating habits used in the model for the velvet bean caterpillar. The dwell time distribution for the VBC is based on experimental data as are the data for the eating habits of the caterpiller. [3, 4, 5] Also shown on Figure

Table 1

INDEPENDENT VARIABLES

Variable Description	Used For	Source For Numerical Values	Data Basis
1. Start time, gen. 1	Pests	Data	Watson, 1916; Greene, 1970
2. No. init. females/acre	Pests	Data	Greene, Kerr, Whitcomb, 1971
3. R. N series	Pests	Data	
4. P(survival) - ea. instar	Pests	Data	P. Lawrence, 1971
5. Avg. dwell time - ea. instar	Pests	Data	
6. σ, dwell time - ea. instar	Pests	Data	Watson, 1916; Greene, 1971
7. P(adult = female)	Pests	Data	
8. Egg laying distribution	Pests	Data	Greene, 1971
9. Leaf area/cat./instar	Pests	Data	Greene, 1971
10. Day of count	Pests	Program	
11. Females forced at gen. end.	Pests	Program	
12. Day of planting	Crop	Data	Greene, 1971
13. Plant growth	Crop	Program	Greene, 1971
14. Leaf area	Crop	Program	Turnipseed, 1972

DEPENDENT VARIABLES

Calculated at Each Day of Count

1. Tot. pop/instar - including eggs
2. Tot. insect population
3. Cumulated defoliation in cm^2
4. Tot. crop leaf area available in cm^2
5. % defoliation

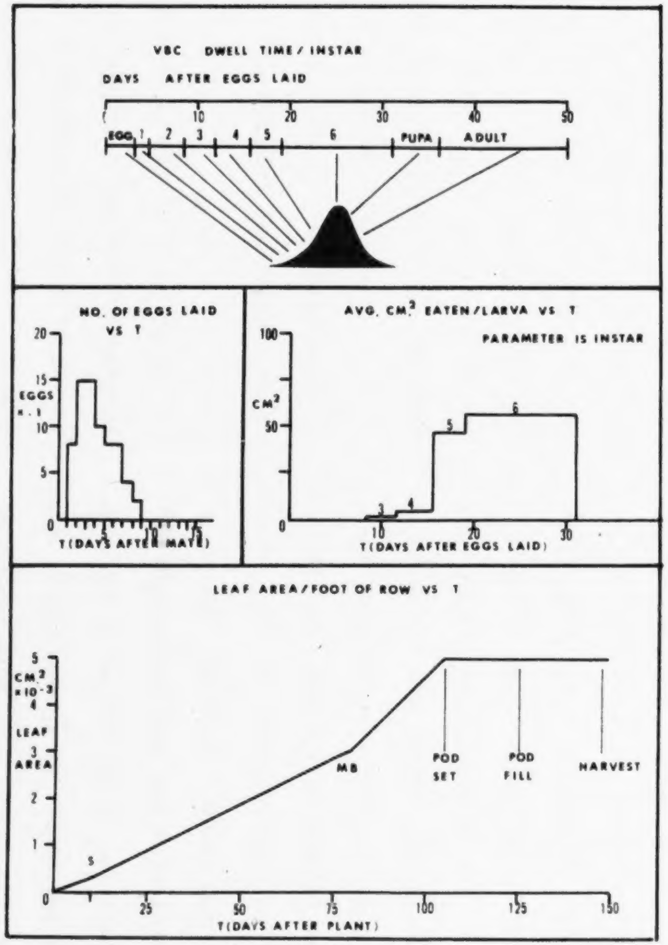


Figure 3

```
IN       RATEC       RNC          RNNC    RMEANC        SIGC
 1     0.7000       667           153    0.0000        0.0000
 2     0.7000       233           497    3.0000        0.3000
 3     0.7000       817           779    1.7000        0.1000
 4     0.7000        19           597    3.6000        0.1000
 5     0.7000        31           673    3.5000        0.1000
 6     0.7000       437           739    3.7000        0.2000
 7     0.7000       913           707    3.5000        0.4000
 8     0.7000       609           219   12.0000        2.0000
 9     0.7000       803           369    5.0000        0.3000
10     0.7000       327           197   14.0000        1.0000
11     0.5000       377           763    0.0000        0.0000
  8   18   33   48   58   66   74   78   80

    START GEN 1 IS   199.0000 DAYS AFTER JAN. 1
ORIG. START FEMALES ARE    60
                                        RANDOM SERIES IS.     4.00
                                        DAY OF PLANTING IS. 127

  0.0      0.16      0.64      3.05      8.25      43.30      62.60

HERE1

HERE2

IMMC=     1

HERE1

HERE2
   1     295.1995
   2     296.1492

IMMC=     2

HERE1

HERE2
   1     340.9487

HERE2
   2     339.0483

IMMC=     2

HERE1

HERE2
   1     388.6477

HERE2

IMMC=     1

HERE4
```

Figure 4

3 is the assumed development for the soybean crop, in leaf area per foot of row in the field.

Figure 4 is a typical computer printout showing the data used in the simulation run and the resultant egg laying dates for females from each generation.

Figure 5 is a typical census printout showing the six census responses enumerated previously.

Preliminary runs with the model confirmed it was following the entomologist's belief that about two generations of the VBC propagated during the time period from the initial moth invasion to the final stage in soybean development where the host crop was sensitive to economic damage caused by defoliation. It was observed that the survival rates at each instar and the differing stochastic dwell-times in each instar combined to produce a mixed response for population as a function of time. That is, the two random variables, the early peak population for each generation and the population in the latter instars including the number of females laying eggs for the next generation, both exhibited large variances from run to run. However, the population counts for time periods when most of the population was in the central instars (roughly instar 3 through instar 5) approached closely EV calculations in these regions. The slope of the population with time in this region was exponential as expected.

This mixed response model therefore does not appear to fit any simple analytical model but does follow the trends predicted by the analytical population models of Watt [6], Pielou [7], Ross [8] and others.

The above details about the population dynamics are apparent in Figure 6 which displays typical population dynamics as predicted by simulation. Slope validation is indicated by the circled points which are calculated EV (overall fraction survival) plotted at weekly intervals, the first plotted point being fraction survival one week after eggs hatch = middle of 3rd instar = $.7^3$ = .343.

Location of the range in time for the second generation population peaks at Figure 6 may be verified by ordinary statistical calculations when it is realized that the start of the second

```
POP. IN INSTAR  2 ON DAY 206 IS              300
POP. IN INSTAR  3 ON DAY 206 IS               70
POP. IN INSTAR  4 ON DAY 206 IS               20
POP. IN INSTAR  5 ON DAY 206 IS                0
POP. IN INSTAR  6 ON DAY 206 IS                0
  P. IN INSTAR  7 ON DAY 206 IS                0
POP. IN INSTAR  8 ON DAY 206 IS                0
POP. IN INSTAR  9 ON DAY 206 IS                0
POP. IN INSTAR 10 ON DAY 206 IS                0
YCT POP ON DAY    206 IS          390
         TOT INS ON DAY   206 IS               90
DAMAGE ON DAY 206 IS          10.09 SQ.CM.

TOTAL LEAF AREA ON DAY 206    39900512.00SQ. CM.

CUM. DAMAGE ON DAY 206 IS     0.00 PERCENTAGE OF LEAF AREA
POP. IN INSTAR  2 ON DAY 213 IS                0
POP. IN INSTAR  3 ON DAY 213 IS               20
POP. IN INSTAR  4 ON DAY 213 IS               60
POP. IN INSTAR  5 ON DAY 213 IS               90
POP. IN INSTAR  6 ON DAY 213 IS                0
POP. IN INSTAR  7 ON DAY 213 IS                0
POP. IN INSTAR  8 ON DAY 213 IS                0
POP. IN INSTAR  9 ON DAY 213 IS                0
POP. IN INSTAR 10 ON DAY 213 IS                0
TOT POP ON DAY    213 IS          170
         TOT INS ON DAY   213 IS              170
DAMAGE ON DAY 213 IS          190.81 SQ. CM.

TOTAL LEAF AREA ON DAY 213    46624160.00SQ. CM.

CUM. DAMAGE ON DAY 213        0.03 PERCENTAGE OF LEAF AREA
POP. IN INSTAR  2 ON DAY 220 IS                0
POP. IN INSTAR  3 ON DAY 220 IS                0
POP. IN INSTAR  4 ON DAY 220 IS                0
POP. IN INSTAR  5 ON DAY 220 IS               10
POP. IN INSTAR  6 ON DAY 220 IS               50
POP. IN INSTAR  7 ON DAY 220 IS               60
POP. IN INSTAR  8 ON DAY 220 IS                0
POP. IN INSTAR  9 ON DAY 220 IS                0
POP. IN INSTAR 10 ON DAY 220 IS                0
TOT POP ON DAY    220 IS          120
         TOT INS ON DAY   220 IS              120
DAMAGE ON DAY 220 IS          1819.31 SQ.CM.

TOTAL LEAF AREA ON DAY 220    53347792.00SQ. CM.

CUM. DAMAGE ON DAY 220 IS     0.20 PERCENTAGE OF LEAF AREA
```

Figure 5

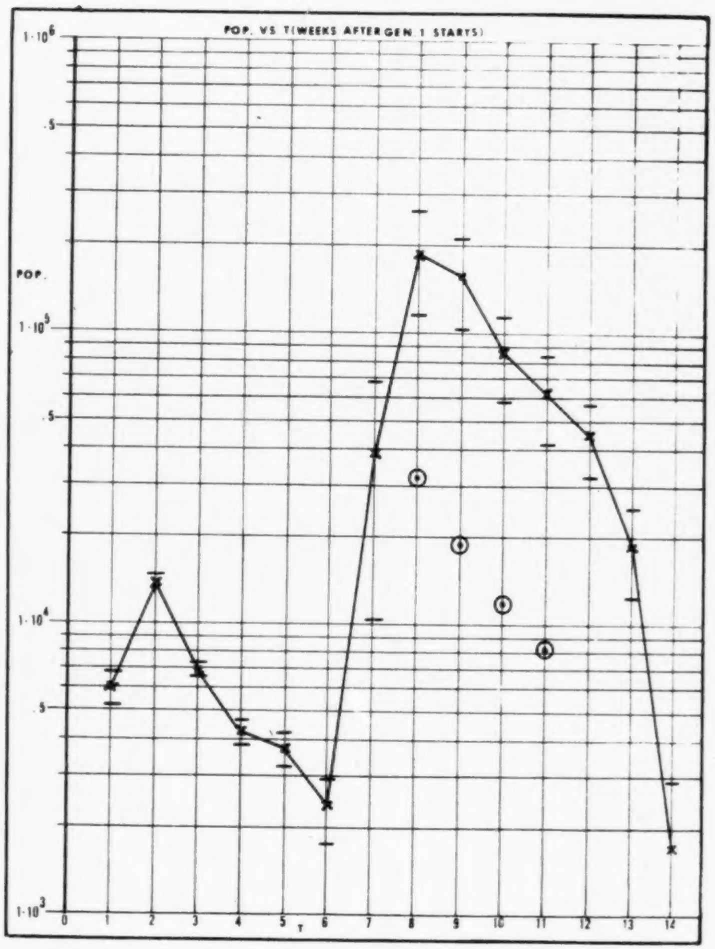


Figure 6

generation is not statistically independent of the first generation. The magnitude of the second generation population peak is a stochastic variate strongly dependent on the number of females that survive the first generation through the egg laying stage. The times at which these lay eggs form the starting conditions for the second generation; the number of survivors determines the second generation population characteristic. Thus the model, as in real life, shows highly volatile, or transient, second generation starting populations; individual second generation population counts can be expected to show significant variance around the average of many simulation encounters.

Validation of results has to date been restricted to the Turing Method both because of the expense involved in time and money and because of the difficulty of obtaining definite experimental data. Population counts in the field are to be taken this summer, 1972. The entomology consultants agree that the results are reasonable and of the correct order of magnitude. One set of population data consisting of three replications in two separate field areas in North Florida was obtained during the 1971 soybean season and conforms to the simulated predicted shape of the average population vs. time. The agreement encourages the belief that the principles used in model construction are valid and that inferences derived from model results merit serious consideration.

V Description of the Experiment

Previous experiments with this model had already determined that the worst possible situation for crop damage occurred for the combination of latest possible planting date for the crop (June 24) and the earliest possible initial invasion date (July 15) for the VBC moth. Note that these two variables are largely uncontrollable and are states of nature, both in the usual sense of these words and in the context of decision making.

Consider the above worst case and rephrase the experimental objective specifically in terms of the following:

$P_j(k)$ = Reduced survival fraction in instar j = $k \cdot Pj$

Pj = Normal survival fraction in instar j k = Fraction of kill induced by any method

j = Instar

G = Generation in which kill is induced

Then define the experimental objectives as follows.

Find a preferred instar, j, in which a fixed fraction kill, k, will minimize the total per cent of defoliation of the soybean crop by the VBC. Find the implications for pest management strategies when the fixed kill is induced in the first generation only, in the second generation only, or is induced in the preferred instar in both the first and second generations. The fixed conditions for the experiment were:

Worst case states of nature

A massive initial VBC moth invasion (60 females/acre)

Experimental responses were:

Population vs. time

Per cent defoliation vs. time

Census counts every seven days

The experimental design for the simulation model was a complete factorial experiment with three factors, k (fraction kill at two levels), j (instar) at three levels and G (generation in which k is induced) at three levels.

Factor levels were chosen as tabulated below:

k = .25, .5

j = 2nd, 4th, 6th instar

G = generation 1 only, generation 2 only, both generations 1 and 2.

Each response was replicated twice, using ordinary then antithetic variate random number series in order to reduce the variance of the average response. It was believed that this experiment was the minimum size permissible, given the expected appreciable variance in the responses. Responses were compared at podset and podfill, critical points in time in the soybean crop development. These points roughly bracket the time period in which the crop is highly sensitive to defoliation effects. Nine sets of randomly selected random number seeds similar to the set of Figure 4 were used as data in order to generate the ordinary and antithetic random variates for the experiment.

Table 2 shows the averages of the replicate responses (average per cent

Table 2

Per Cent Defoliation at Podset

	k = .25			k = .5		
	j^2	j^4	j^6	j^2	j^4	j^6
G_1	30.8	37.0	34.5	22.2	32.1	21.6
G_2	33.0	30.7	44.4	22.6	25.9	39.5
$G_{1,2}$	23.2	25.7	37.5	12.0	13.7	22.3

Per Cent Defoliation at Podfill

	k = .25			k = .5		
	j^2	j^4	j^6	j^2	j^4	j^6
G_1	39.3	46.7	43.6	27.2	41.2	26.8
G_2	43.4	38.8	53.2	28.3	31.9	45.6
$G_{1,2}$	27.6	31.8	46.1	14.8	17.3	26.1

k = fraction kill

G = generation in which k is induced

j = instar in which k is induced

Table 3

ANOVA - At Potset

Source of Variation	Degrees of Freedom	Sums of Squares	Mean Squares	F	F_{sig}
j	2	532.08	266.04	2.9	$F_{.05}=3.32$
G	2	672.06	336.03	3.7	$F_{.05}=3.32$*
k	1	800.89	800.89	8.72	$F_{.01}=7.56$**
Error	30	2752.5	91.8		

Table 4

ANOVA - At Podfill

Source of Variation	Degrees of Freedom	Sums of Squares	Mean Squares	F	F_{sig}
j	2	617.9	309.0	2.33	$F_{.1}=2.52$
G	2	1098.8	549.4	4.13	$F_{.05}=3.37$*
k	1	1366.5	1366.5	10.3	$F_{.01}=7.72$**
jxG	4	634.3	158.6	1.2	not sig.
Error	26	3455.8	133.		

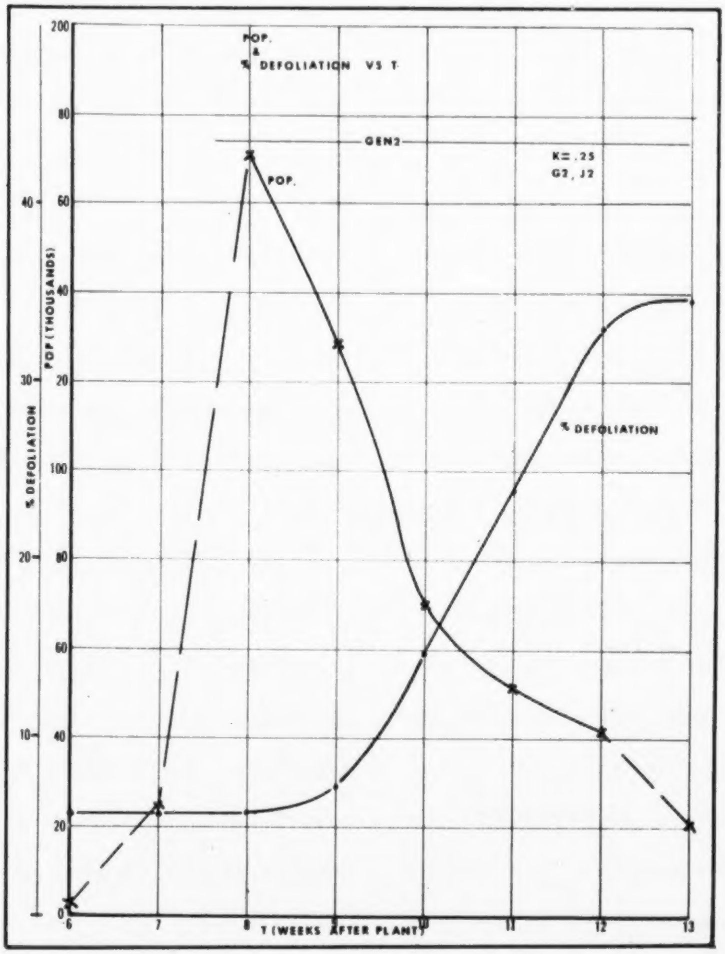

POP.
&
% DEFOLIATION VS T
GEN2
K= .25
G2, J2
POP.
% DEFOLIATION
% DEFOLIATION
POP (THOUSANDS)
T (WEEKS AFTER PLANT)
200
80
60
40
20
100
80
60
40
20
0
40
30
20
10
6
7
8
10
11
12
13

Figure 7

defoliation) for each cell of the experimental design at both podset (Oct. 10) and podfill (Oct. 30).

Figure 7 presents superimposed results for the variables, insect population and cumulative damage, (per cent defoliation) vs. time. The time phasing between the two response variables as well as the complex functional relationships that must relate them is clearly indicated.

Figure 6 is provided to indicate the order of magnitude of the population variances generated by the model. The data of Figure 6 were obtained in previous experiments with this model and are the results of six replicate simulations each using a different set of random number series. The 95% confidence limits are indicated by the horizontal lines bracketing the averages for the six runs. It can be noted that the spread of the 95% confidence limits is highly time dependent.

VI. Analysis and Discussion of Results

Analyses of variance for the results of the experiment are shown in Table 3 for podset and Table 4 for podfill. Both sets show consistent results implying that similar relations between k, j, and G exist at these two points in time. There is no reason to believe the relations change at points in time between these end points.

As expected, fraction kill, k, is a highly significant factor of the experiment. Even with the minimal data of these runs, k was significant at the 1% level. The generation at which kill takes place, G, the next most important factor was significant at the 5% level. The instar at which k occurred, j, was significant at only the 10% level. It is believed that additional replications would show this factor to have a greater effect.

The experimental data were further analyzed at podfill by burying the effects of all levels of the instar and kill factors in order to determine which of the generation factor levels were significantly different. The significance of the difference between generation means was tested by the t test described below.

Mean response

G_1 kill	G_2 kill	$G_{1,2}$ kill
u1	u2	u both
37.48%	40.23%	27.38%

u = average response

n = 12 responses/column

d. of f. = 22 for any 2 column comparisons

$\hat{s}$ = estimated $\sigma_e = \sqrt{133}$ (from ANOVA)

$t = (\bar{y}_c - \bar{y}_j)/\hat{s}\ \sqrt{2/n}$

$= (\bar{y}_c - \bar{y}_j)/4.71$

where $\bar{y}_c$ = control column mean

$\bar{y}_j$ = test column mean

t_{crit} for $\alpha = .05$, d. of f. = 22: $t_c = 2.074$

Test 1

H_o: $u_1 = u$ both

$t = (27.4 - 37.4)/4.71 = -2.12$

Therefore reject H_o and accept

H_1: $u_1 \neq u$ both at 95% C.L.

Therefore % defoliation for kill at both ($G_{1,2}$) is significantly lower than that for G_1 only.

Test 2

H_o: $u_2 = u$ both

$t = (27.4 - 40.2)/4.71 = -2.72$

Therefore, reject H_o and accept

H_1: $u_2 \neq u$ both at 95% C.L.

Thus per cent defoliation for kill at $G_{1,2}$ is also significantly high- than that for G_2 only.

Test 3

H_o: $u_1 = u_2$

By inspection it can be seen that there is no significant difference between per cent defoliation for G_1 and G_2.

This series of tests appears to indicate that a one time kill is equally effective at either generation one or generation two. As anticipated, a kill applied in both generations results in significantly lower defoliation than a one time kill.

A similar set of tests was made by burying the effects of all levels of the kill and generation factors and comparing mean response for the instars to detect significant instar levels.

Mean response

j = 2	j = 4	j = 6
uj2	uj4	uj6
30.12%	34.71%	40.25%

u = average response

n = 12 responses/column

d. of f. = 22 for any 2 column comparison

t_{crit} for $\alpha = .05$, d. of f. = 22: $t_c = 2.074$

Test 4

H_o: uj4 = uj2

$t = (30.1 - 34.4)/4.71 = -.98$

Therefore accept H_o at 95% C.L.

Therefore there is no significant difference in the effects of instars 2 and 4 upon per cent defoliation.

Test 5

H_o: uj6 = uj2

$t = (30.1 - 40.2)/4.71 = -2.14$

Therefore, reject H_o and accept H_1: uj6 $\neq$ uj2 at 95% C.L.

Therefore per cent defoliation caused by a fixed k at instar 6 is significantly higher than that for the same k at instar 2 or 4.

In summary, the two sets of tests reported above indicate:

A. A one time fixed-fraction kill, induced at the same instar in either the first or second generation, is equally effective in controlling per cent defoliation that occurs during the second generation.

B. Early instars (2 through 4) are preferred for application of the induced kill.

C. Kills at both generations, at the same instar, are more effective in minimizing per cent defoliation than one generation kills.

Reference to Figure 6 shows that the second generation population of the VBC is about an order of magnitude larger than that of the first generation. Assume, as a first approximation, that the amount of the factor inducing the kill is proportional to the number of caterpillars to be killed. The fixed fraction kill at the second generation will then require an order of magnitude greater amount of the kill factor, be it insecticide, predator, parasite, etc., than that for the same fraction kill at the first generation. Thus, result A above must be modified to indicate that generation one is preferred in view of the original objective to keep the amount of kill factor at a minimum.

VIII - Conclusions

The above analyses suggest the following biological control strategies should be investigated in the real life situation.

1 - Apply kill factor early in the first generation. This implies a payoff exists for emphasizing early detection of the first generation larvae.

2 - A period of about two weeks is available during each generation's life cycle (Result B above) in which the strategy of 1 above can be used. That is make haste slowly; time is available for careful planning for the most useful strategy.

3 - If the early instars of the first generation are undetected, concentrate on the early instars of the second generation. (The last instars in any generation of the VBC are nearly impossible to kill, per the practicing entomologists). The penalty for having to use this approach is a large increase in the amount of kill factor employed during the favorable two week period.

One has indeed been able to identify viable biological strategies by analysis of the results of a designed experiment for the simulation model. The relative factor effects of instar (j), fixed faction kill (k), and generation (G) at which the kill is induced have been evaluated.

The results of this experiment suggest that future experiments should include fraction kills at the egg stage. Should the egg fraction kill be signicantly effective, experimental efforts on pest controls by specific parasites and predators should be increased. It is anticipated that future research will extend the model design to include environmental effects, pest-predator-parasite-fungi relations, economic damage caused by multiple pests, and consideration of pollution or poisoning effects of alternate management strategies.

IX - Technical Note

Average cost/simulation run = $15. (IBM 360/65).

X - Acknowledgements

This work could not have been accomplished without the cooperation, advice, criticism, and the continual education of a management scientist in biological principles by the University of Florida's consulting entomologists, Dr. Gerald

Greene, Dr. Stratton H Kerr, and Dr. Willard H. Whitcomb. Many thanks, gentlemen.

BIBILOGRAPHY

1. Majumder, S. K., Vegetarianism: Fad, Faith, or Fact? American Scientist, Vol. 60, No. 2, March-April 1972, p. 178.
2. Hillier, F. S., Lieberman, G. H., Introduction to Operations Research, Holden-Day, San Francisco 1968, pp. 228-229.
3. Drake, C J., The Southern Green Stink Bug in Florida, The Quarterly Bulletin, State Plant Board of Florida, Vol. IV, No. 3, April '20.
4. Watson, J. R., Life History of the Velvet Bean Caterpillar, Journal of Economic Entomology, Vol. 9, December '16.
5. Greene, G., Private Communication, 1971-72.
6. Watt, K. E. F., Mathematical Population Models for Five Agricultural Crop Pests, Entomology Soc. Can. Memoirs, No. 32, 1963.
7. Pielou, E. C., An Introduction to Mathematical Ecology, New York, Wiley-Interscience, 1969, p. 1-65.
8. Ross, S. M., Applied Probability Models With Optimization Applications, Sas Francisco, Molden-Day pp. 18, 55.

Session 2: Simulation Methodology I
Chairman: Mark Garman, University of California

This session treats methods and techniques in the utilization of simulation models. Applications methodology is explored in the first paper, which employs simulation as a tool to validate an analysis via the discussion of multiple-sequence random number generation. Technique development is the subject of the third paper, which treats the implementation of "wait-until" time flow mechanisms.

Papers
"Use of Simulation to Test the Validity and Sensitivity of an Analytical Model"
Prosper Bernard, The Provincial Bank of Canada

"Multiple Sequence Random Number Generators"
Joe H. Mize, Oklahoma State University

"A'Wait-Until' Algorithm for General Purpose Simulation Languages"
Jean G. Vaucher, Universite de Montreal

Discussants
Don Gaver, Naval Postgraduate School
Kenneth Siler, University of California

USE OF SIMULATION TO TEST THE VALIDITY AND SENSITIVITY OF AN ANALYTICAL MODEL

PROSPER M. BERNARD
The Provincial Bank of Canada

ABSTRACT

The purpose of the research reported here was to test formally the validity of some assumptions made in solving models by analytical techniques and to test the sensitivity of the system to the arrival distribution. The presentation refers to a simplified real-time model. The analytical solution in the literature is to derive shown results for each stage of the system and add them up to obtain the behavior of the system. Assumptions must then be made at each stage. Simulation has been used to solve the same system in terms of the same measure of efficiency, i.e. the response time. However, the system is solved as a whole, the output from one stage becoming the input to the second stage. Confidence limits have been obtained for the response time in order to test the results obtained from analytical techniques. Simulation has also been used to test the sensitivity of the system to a change in the arrival distribution. Using analysis of variance, the effect of the arrival pattern and of the interaction is determined.

Introduction

The purpose of this presentation is to show how one can use simulation to verify formally some assumptions made in solving a model by analytical techniques and to test the sensitivity of that model to a departure from the classical

"Poisson" arrival assumption. The method used in this paper is to describe the model, which has been published, and to discuss an analytical solution. As will be seen in detail later, the procedure for solving this multi-stage model of queues in series is first to solve each stage and then to add up the results. This implies making assumptions at each stage, and this procedure may introduce errors. For this reason, the simulation will be used to study the system as a whole without having to make intermediate assumptions. Independent simulation experiments will be made in order to obtain confidence limits on the mean response time.

Most queuing models assume a Poisson arrival distribution and no solution is offered for a departure from the "Poisson" assumption. It might be easier to determine in advance the effect of the arrival distribution than to make field studies to make sure that it is in fact Poisson. Even if one could determine the exact shape of the distribution and it were found not to be the Poisson distribution, there is no solution available. For this reason the author will test the sensitivity of the model by repeating simulation experiments with distributions as far apart as Poisson, Normal, and Uniform.

Model to be tested

In this real-time inquiry model there are a finite number of customers linked to a central processor via common carrier lines and a terminal. A graphical representation of the model is given in figure 1.

The following assumptions have been made:

1. The arrival distribution at each terminal is Poisson with mean λ/m where m is the number of terminals.
2. The service time of the central processor is distributed according to an Erlang-2 distribution.
3. The service time for each terminal, i. e. the key-in time and print-out, are uniformly distributed.
4. The queue discipline is First-in-First-out.

The real-time inquiry model considered in this presentation could be seen as a machine interference model where there are external arrivals to each machine or, in this case, to each terminal. Customers arrive at each of the m terminals randomly with mean arrival rate λ/m. The total response time consists of the waiting-time for the terminal, the service-time of the terminal (key-in), the waiting for the CPU, the service-time in CPU and the service-time of the terminal (print-out). The system is considered busy during the whole period and can accept a new customer only after the terminal processing of the previous customer.

The reader will see the similarity between this model and the classical "machine interference" model. The whole model can be seen as a multi-stage model where the middle stage is a machine interference model. This latter model has been well described and analyzed in the literature.

In the machine interference model there are a finite number of machines or sources assigned to one serviceman or service station. The machine is either "up" or "down". When the machine goes down it joins the queue for service. The machine gets immediate service or waits for service depending on the availability of the single repairman.

In the machine interference model the following assumptions are usually made:

1. The service time is exponentially distributed with mean $\bar{T}s$.
2. The "up" time for each machine is exponentially distributed with mean time $\bar{T}a$.

These assumptions are sometimes referred to as the worst-case conditions. The ratio of the two times is defined as the "service ratio" where

$$Z = \frac{\bar{T}a}{\bar{T}s}.$$

The "Server Utilization" denoted as $r_{m(z)}$ can be obtained for each number of machines m and for each value of z as follows. Let the probability P_0 represent the fraction of time when there are zero machines in the service queue, and the serviceman is idle. Thus, $1 - P_0$ may be interpreted as the fraction of time the serviceman is busy, that is:

$$\text{Server Utilization} = 1-P_0 = 1 - \frac{e^{-z}\frac{z^m}{m!}}{e^{-z}\sum_{j=0}^{m}\frac{z^j}{j!}} = r_{m(z)}.$$

Analytical solution

The model has been given not a rigorous solution, but an approximation, where the results of all stages are added up to make the total response time. The following solution has been proposed[1].

From the user's point of view, the system is in use when he starts keying the inquiry so that it could be considered as a service-station or a "black box" in which service time Tp is:

$$Tp = Tu + Tw + Ts + To \qquad 1$$

where

Tu = Time to key-in and transmit the message

Tw = Waiting time to access CPU

Ts = Time for service in the CPU

To = Time to transmit and print the result

Tp = Total service time

Tq = Total response time = waiting for the system + Tp.

The model is represented in figure 1.

Since the overall system can be considered to be a single service station where the expected service-time is $\overline{T}p$, the system utilization by a user can be defined as:

$$\rho = (\lambda/m) \;/\; (1/\overline{T}p) = \frac{\lambda}{m} \cdot \overline{T}p \qquad 2$$

The response time for a single server model with random arrivals and an arbitrary service distribution has been obtained by Pollaczek and simplified by Khintchine[2]. The general formula known as Pollaczek-Khintchine uses only the first two moments of the service distribution and can be transformed by algebraic modifications to:

$$Tq = \frac{Tp}{1-\rho}\left[1 - \rho/2\left(1 - \frac{\sigma_p^{2}}{Tp^{2}}\right)\right] \qquad 3$$

The total response-time has been obtained in terms of Tp, the service-time when the overall system is assumed to be a service-station. The only further information necessary to obtain the expected value of Tp is the expected waiting time to the CPU itself.

By applying the machine interference results for the middle subsystem where the machines or terminals are queuing the CPU, one can obtain the remaining values. The server's (CPU) utilization R(m)Z can be found from the graph in figure 2. However, the service ratio z is not known, since it depends on the external arrival rate. This ratio can be determined in the following manner. Since all customers arriving at the various machines must eventually go through the service queue, the utilization of the serviceman can be calculated, independently from z, to be

$$r_m(z) = \lambda \cdot \overline{T}_s \qquad 4$$

Z $\overline{T}s = \overline{T}a$ and $r_m(z) = \lambda\overline{T}s$ can be substituted in the general machine interference formula

$$E(\text{time between breakdowns}) = \frac{mts}{r_m(z)} - ta.$$

The following results are thus obtained.

$$Tw + Ts = M/\lambda - Z\; Ts \quad \text{if} \quad \frac{\overline{T}w}{Ts} > 1. \qquad 5$$

However, for $\frac{\overline{T}w}{Ts} < 1$, the simple queuing time formula may be used as a good approximation.

$$Tw + Ts = \frac{Ts}{1 - \frac{(m-1)}{m}\lambda Ts} \quad \text{if} \quad \frac{Tw}{\overline{T}s} < 1 \qquad 6$$

where Z is the service ratio in the machine interference model.

Example:

The behavior of the model can be shown better in terms of an example. In this example there are 20 terminals connected to the CPU. The key-in time is uniformly distributed between

5 and 15 and the print-out of a message is also uniformly distributed between 2 and 7. The computer processing time is assumed to be an Erlang-2 distribution with a mean of 2 seconds. Although the CPU processing time is not exponentially distributed, the machine interference formulae are used as an approximation.

In order to obtain numerical values for the expected response-time and for the other components one may proceed as follows:

CPU utilization is determined from equation

$$r_{20}(z) = \lambda \frac{\text{Inquiries}}{\text{sec}} \times \frac{2 \text{ sec}}{\text{inquiry}}$$

From this equation the service ratio z is determined and can then be substituted in equations 5 and 6.

$$\overline{T}_w + \overline{T}_s = \frac{2}{1-0,95 \times 2 \times \lambda} \text{ if } \frac{Tw}{Ts} < 1$$

or

$$\frac{20}{\lambda} - 2z, \text{ if } \frac{Tw}{Ts} > 1.$$

The overall inquiry service time found from equation 1 is then given as

$$Tp = 10 + Tw + Ts + 5.$$

The inquiry service variance is likewise given as the sum of variances

$$\sigma_p^2 = \frac{(15-5)^2}{} + Tw^2 + \frac{2}{2}^2 + \frac{(7-3)^2}{12} = 11,7 + \overline{T}w^2$$

where σw^2 is approximated by Tw^2.

Terminal utilization is given from equation 2.

$$\rho \frac{\lambda}{20} = Tp.$$

Finally, the inquiry response time is determined from the queuing time formula of equation 3. The whole set of calculations is summarized in Table 1.

With 20 terminals the system can accept 0,4 inquiries per second or 1,2 inquiries per minute per terminal. The response time is 30 seconds with CPU utilization of 80%. Beyond this point the response time increases at an increasing rate as shown in figure 5.

The reader will notice that assumptions have been made above concerning the arrival distribution at the second stage of the system. Burke has shown that the output of one queue with Poisson input is also Poisson[3]. However, the general procedure for obtaining the distribution of the output of queues in parallel and series is difficult to obtain analytically. In this analytical model, assumptions have to be made at each stage. It is assumed that the input to the second stage is Poisson as weel as the input to the system.

In solving the model, assumptions were made and approximations were used. The model was analytically solved by studying each of the subsystems and adding up the results. Intermediate assumptions were made concerning each subsystem. The middle stage was approximated by the machine interference model where the "down" times are exponentially distributed. Even in that middle stage, the usual queuing theory formulae for arrivals from an infinite population were used where the ratio of the waiting time over the service-time at the CPU was small.

Like any abstraction, the model represents only some aspects of the real world. It is important, therefore, to know the effect of assuming or neglecting certain parameters. The model is based on the "Poisson" arrival assumption and no solution is given for a non-Poisson arrival. If may be asked what would happen if the arrival distribution was not really Poisson. To explore this, the two following hypotheses will be tested.

Hypotheses

A. The expected total response time is properly obtained from the analytical solution.

B. The arrival distribution has no effect on the expected total response time of the simulation.

Simulation results

Simulation runs using GPSS were made to test the above hypotheses. The simulation runs are made for the same values as in the example given above. Th program generates Poisson arrivals randomly distributed to any of the 20 terminals. A queue is formed in front of each terminal. The units go through the stages of the model illustrated in figure 1. In the simulation, there is no need to make assumptions at each stage since the program will take as input in stage 2 the output of stage 1. Measurements are made only at the end, for the customer is interested in knowing the response time of the overall system rather than the waiting at each stage.

The First Hypothesis. The system was studied at eight different arrival rates corresponding to the rates shown in table 1. In order to arrive at a confidence interval at each point of interest one must have a sample of independent observations. However, data generated by simulation are autocorrelated. If we assume the absence of autocorrelation we may underestimate the variances or we may take a too small sample. The variance of autocorrelated data is not related to the population by the simple expression

$$\sigma^2_{\bar{x}} = \sigma^2/n$$

but by

$$\sigma^2_{\bar{x}} = \sigma^2/n + k$$

where k is a positive number. In order to avoid the problem of autocorrelation, 12 independent runs were made at each of the 8 arrival rates being studied. The mean of each run was used. In each run a transient period of 50 arriving units was discarded and 100 steady state units were recorded. By making exploratory runs it was found that the steady-state was reached well before 50 units had arrived. The mean of the response time for each is approximately normally distributed and a confidence interval can be calculated.

It is not possible to show here the results of so many runs. However, the expected response time for each run was recorded and entered in the first column of table 2.

The output of the simulation runs are compared to the analytical results in table 3. Although our interest here is in the expected total response time, table 2 also shows the utilization values, the waiting times and the length of the queue. However, the response time has not been obtained by adding up the different items but by measuring the difference of time between the arrival and the departure of a unit. The other values are presented to help identify the area of great differences, as will be discussed later. Both response times are graphed in figure 6. It will be noticed that there is a major difference in the results when the arrival rate approaches 1.0 unit per second or when the CPU utilization approaches 1.

The analytical results imply that the mean response time curve shown on figure 1 approaches asymptotically a vertical line at an arrival rate below 0.5 inquiry per second. The simulation results show that the expected response time curve approaches asymptotically a vertical line at 0.75 inquiry per second.

It should be noted here that the results of the simulation model were not obtained starting from empty system. This would have produced results further away from the analytical results. Instead, the model was run until steady state had been reached. Only at that point were the statistics accumulated.

As one will recall, the analytical solution was obtained by adding the waiting time and the service time at each stage. One can see, by looking at table 3, that the element that varies the most between the analytical and the simulation results is the waiting time at the CPU itself. This corresponds to the waiting time at the stage that was approximated by the machine interference model. It is interesting to note that in the area where the ratio of the average waiting time to the CPU over the average service time was less than one (i.e.) $\frac{\overline{T}w}{Ts} \angle 1$). the results of the analytical solution and of the simulation are statistically the same. In this region the machine interference model was not used.

Confidence intervals for the total response time were calculated using Student t Statistics for the 8 arrival rates of interest at 99% level and 95% level. The results are summarized in table 4 and plotted in figure 4. It can easily be seen that for an arrival rate of over .45 inquiry per second , the analytical results are well outside the 99% confidence interval.

It is easy to conclude that the results fail to confirm the first hypothesis. The response time does not increase as fast as suggested by the analytical techniques. The reason for this may be that the input to the second stage of the model, *i. e.* the CPU, is not exponentially distributed, as has been assumed to fit the machine interference model. The later model assumed exponentially distributed "down" times on each of the 20 terminals. However, since a queue is formed in front of each terminal followed by a uniform service time, the equivalent of the "down" times are not necessarily exponentially distributed. The reader will notice that in the simulation runs there is no need to make assumptions at the second and each subsequent stage as in the analytical solution.

Testing the Second Hypothesis. An important factor often mentioned in the literature is that if one wants to use the arrival rate, a special study should be undertaken to make sure that it is Poisson. In this presentation the author uses another approach, *i. e.* sensitivity analysis of the assumptions. One would ask *What if* the assumptions are not true? Should we investigate the real arrival pattern?

It is obvious that there is no need to investigate the exact value of a factor *if* this factor has no effect on the system. The author's approach is therefore, to determine in advance the effect on the system of the mean of the arrival distribution. In order to do this, the author repeated the experiments described above for two other distinctive arrival patterns, *i. e.* *normal* and *uniform* arrival. The means of the three distributions are the same. The uniform distribution varies from 0 to 2 $\overline{X}$ where $\overline{X}$ is the mean. The standard deviation used for the normal distribution is in this case $\overline{X}/5$. In each case, as before, 12 runs of 100 steady state arrival were made for each of the 8 levels of interest. The results obtained are summarized in table 2. There are 288 runs of 100 observations, *i. e.* 28,800 observations in all, not including the transient period. We also have enough information to test the interaction since each cell in the table has 12 observations.

An analysis of variance was made to determine the effect of the treatment (arrival distribution) on the response time. The arrival distributions and the arrival means are both fixed factors. There are 3 levels for the first and 8 levels for the second factor and 12 independent observations in each cell as shown in table 2.

The arrival means have an obvious effect on the mean response time. However, here one is interested in testing the effect of the arrival distribution and the interaction effect. The results of the analysis of variance are shown in table 5. The surprising conclusion is that the arrival distribution has no significant effect on the total response time (even at a low 73% confidence level) and that the interaction, i. e. the combined effect of arrival rate and arrival pattern, is almost non-existent. We do not reject the second hypothesis.

This conclusion implies that there is no need to search for the true arrival pattern since it has little effect. Even if the arrival to the system exhibits a departure from the Poisson assumption the expected response time seems not to be affected.

Conclusion

In this paper the author has reviewed and tested by simulation a simplified real-time model. He has shown that the system does not behave as prescribed by the analytical solution and has demonstrated the original conclusion that the arrival distribution has little effect on the expected total response time. However, a more extensive study made by the author on many other models showed that many models are sensitive to a departure from the Poisson arrival.[5] Simulation has been used successfully to arrive at this conclusion because there is no need for assumptions to be made at each stage as in the analytical solution. Furthermore, a simulation model, once running, can easily be changed to analyse the assumptions at the first stage. It is obviously less expensive to run sensitivity analysis on the assumptions than to field-test them. Sensitivity analysis could be performed on any of the other parameters of the model. However, simulation experiments being very expensive, the author selected to allocate limited resources to the analysis of the classical queuing theory assumption, i. e. the Poisson arrival distribution, in a simple model.

Biography

Prosper M. Bernard holds a B.A. from University of Montreal, a B.Sc. from McGill University, and an MBA from St. John's University New York. The degree of Doctor of Philosophy is expected soon from the City University of New York. He is also a CDP holder. Mr. Bernard held teaching positions both in Quebec and in New York. He also has a long experience in the information processing field as a programmer, systems analyst, manager, and management consultant. He is presently with the Provincial Bank of Canada as an internal consultant.

Acknowledgment

The author would like to thank Dr. Lloyd Rosenberg of the City University of New York for his helpful guidance during his research. The author is also grateful to Maritime Overseas Corporation and Israel Discount Bank in New York for the free use of their computer in doing this research. The Provincial Bank of Canada has also being instrumental in making this presentation possible.

References

1. Analysis of Some Queuing Models in Real-Time Systems, (New York: IBM, Ref.: F20-0007-1, 1966), p. 44-46.
2. T. L. Saaty, Elements of Queuing Theory, New York: McGraw Hill, . 1961) p. 40.
3. P. J. Burke, "The Output of a Queuing System", Operations Research, Vol. 4, December 1956, pp. 699-704.
4. Prosper M. Bernard, "Use of Computer Simulation to Test the Validity and Sensitivity of Real-Time and Time-Sharing Queuing Models", PhD Thesis, City University of New York, 1973.

Table 1

Example of the results obtained from the analytical solution

CPU Utilization R_{20}	Arrival rate	Service ratio Z	Waiting time CPU $\overline{T}_w$	Terminal utilization	Total time in system $\overline{T}_q$
0.1	0.05	200	0.21	0.04	17.5
0.2	0.10	98	0.47	0.09	18.5
0.4	0.20	48	41.20	0.18	20.2
0.6	0.30	31	22.65	0.30	24.2
0.8	0.40	21	64.80	0.44	31.2
0.9	0.45	17	67.30	0.55	41.0
0.98	0.49	13	212.5	0.72	75.0

TABLE 2

SIMULATION

EXPECTED RESPONSE TIME

MEAN ARRIVAL RATE SEC.	ARRIVAL DISTRIBUTION		
	POISSON	UNIFORM	NORMAL
0.10	18.7 17.3 18.4 17.6 17.5 18.0 18.5 18.3 18.3 17.6 19.2 17.8	17.9 18.8 18.2 18.1 18.7 17.6 17.3 17.9 18.3 17.9 17.0 18.3	17.5 17.2 17.7 17.9 16.6 17.4 17.3 17.3 18.2 18.4 17.9 17.7
0.20	20.9 21.6 19.4 18.6 18.7 19.8 19.8 22.4 20.9 20.2 18.7 19.6	18.8 19.7 19.4 18.8 20.6 20.8 18.3 20.3 18.0 20.1 19.0 21.3	19.1 17.9 18.6 17.3 18.9 18.9 18.4 19.0 19.5 18.6 18.9 19.7
0.30	22.9 23.3 25.8 21.8 20.4 25.1 22.5 27.1 23.1 20.2 21.0 22.4	22.4 21.6 19.8 20.5 18.9 27.3 22.6 22.4 23.3 20.8 19.9 21.6	22.4 22.9 20.4 20.3 20.8 20.5 21.3 22.3 21.2 23.6 21.8 22.8
0.40	44.1 32.4 27.2 26.4 28.2 21.0 23.1 23.1 28.1 29.5 26.1 33.3	33.5 24.5 20.9 26.2 [illegible] 36.2 24.5 27.2 [illegible] 26.4 32.7 37.9	27.1 24.3 22.1 23.1 23.9 27.2 26.0 24.6 21.5 27.3 26.7 27.8
0.45	41.6 27.7 32.1 43.5 31.2 27.1 37.8 38.4 31.3 24.4 27.7 29.3	32.1 29.7 26.6 32.5 29.1 27.3 25.4 25.0 50.0 32.4 29.8 29.6	32.7 26.6 26.9 29.0 30.3 31.8 23.7 25.6 26.1 30.2 29.6 34.2
0.50	31.9 33.1 57.3 28.9 39.1 37.9 30.9 41.8 59.6 39.7 59.1 70.2	38.3 29.3 46.4 37.4 43.8 39.3 33.7 44.8 58.6 40.6 32.4 33.7	40.7 32.5 32.2 29.0 32.3 46.9 41.0 28.2 31.3 39.6 34.3 39.4
0.555	47.5 39.8 35.2 61.7 40.6 29.4 36.2 25.4 90.6 54.7 37.4 56.5	55.3 33.1 41.1 43.2 61.5 46.8 40.2 54.0 51.2 48.3 45.6 52.2	34.4 43.7 34.7 60.0 39.1 60.6 55.2 41.0 39.5 45.3 59.7 62.1
0.566	65.6106.0 43.7 49.2 95.8 91.1 68.2 57.2 71.2 74.9 63.9 71.5	57.3 65.9 83.4 73.9 64.2 72.5 56.2 80.8 77.8 51.4 45.6 56.7	69.4 45.5 76.6 83.8 76.2 70.0 67.8 69.4 64.1 78.2 51.0 71.3

NOTE: EACH VALUE SHOWN IN THIS TABLE IS THE AVERAGE OF OA RUN MADE OF 100 NON TRANSIENT OBSERVATIONS.

THERE ARE 285 RUNS WITH DIFFERENT RANDOM NUMBER SEQUENCES.

TABLE 3

ANALYTICAL RESULTS VS SIMULATION RESULTS

INTER ARRIVAL TIME (SEC)		ARRIVAL RATE		UTILIZATION OF CPU		WAITING FOR CPU		TERMINAL UTILIZ.		RESPONSE TIME		RESPONSE TIME		LENGTH OF AVERAGE CPU		QUEUE CPU MAXIMUM
ANAL.	SIM.	ANAL.	SIM	ANAL.	SIM.	ANAL.	SIMU.	ANAL.	SIM.	EXCL. ANAL.	TERM SIM.	INCL. ANAL.	TERM. SIM.	ANAL.	SIM.	SIM.
10	10	0.10	0.10	0.2	0.197	0.47	0.324	0.09	0.083	17.5	17.25	18.5	18.109	0.047	0.035	2.0
5.	5	0.2	0.2	0.4	0.395	1.20	0.928	0.18	0.161	18.2	18.48	20.2	20.012	0.240	0.194	6.0
3.33	3.33	0.3	0.3	0.6	0.606	2.65	1.749	0.30	0.321	19.7	18.50	24.2	22.915	0.795	0.518	7.0
2.5	2.5	0.4	0.4	0.8	0.783	4.20	4.017	0.44	0.475	21.8	22.67	31.2	28.557	1.90	1.771	10.0
2.22	2.22	0.45	0.45	0.9	0.895	7.30	5.563	0.55	0.428	24.3	24.00	41.0	32.724	19.0	2.578	10.0
2.0	2.0	0.5	0.5	1.0	0.961	38.0	8.493	1.0	0.708	55.0	LARGE	LARGE	43.793	LARGE	4.281	13.0
1.8	1.8	0.555	0.555	1.0	0.967	LARGE	8.512	1.0	0.916	LARGE-	LARGE	LARGE	46.206	LARGE	4.408	14.0
1.5	1.5	0.666	0.666	1.0	0.999	LARGE	5.601	1.0	0.965	LARGE	LARGE	LARGE	71.010	LARGE	8.516	15.0

ANAL. : RESULTS OF ANALYTICAL SOLUTION
SIM: SIMULATION RESULTS
LARGE: MEANS THAT THE VALUE APPROACHES INFINITY

TABLE 4

CONFIDENCE LIMITS FOR RESPONSE TIME

ARRIVAL		UTILIZATION OF CPU		TOTAL TIME SPENT IN SYSTEM		*RESPONSE	CONFIDENCE INTERVAL FOR POP. MEAN	
TIME	RATE	ANAL.	SIMUL. * MEAN OF MEANS	ANAL.	SIMUL * MEAN OF MEANS	STD. ERR. OF MEAN	95% LEVEL	99% LEVEL
10	110	.2	0.197	18. 5	18.139	0.162857	17.7429 - 18.6042	17.5958 - 18.6042
5	.2	.4	0.395	20.2	20.312	0.326897	19.4145 - 20.8522	19.1213 - 21.1454
3, 33	.3	.6	0.606	24.2	22.915	0.614245	21.6160 - 24.3174	21.0651 - 24.8693
2, 5	.4	.8	0.783	31.2	28.557	1.76127	24.6770 - 32.4230	23.0974 - 34.0026
2.22	.45	.9	0.895	41.0	32.724	1.79238	28.7170 - 36.5997 ***	27.1094 - 38.2073
2, 0	.50	1.0	0.961	LARGE **	43.793	2.52822	28.8989 - 40.0178 ***	26.6314 - 42.2853
1, 8	.555	1.0	0.967	LARGE **	46.206	5.13776	34.9023 - 57.4977 ***	30.2943 - 62.1059
1.5	.660	1.0	0.999	LARGE **	71.010	5.41217	58.9572 - 82.7595 ***	54.1031 - 87.6136

* THE VALUE SHOWN IS THE AVERAGE VALUE OF 12 RUNS. EACH RUN HAS 100 STEADY STATE OBSERVATIONS.

** APPROACHES INFINITY AS UTILIZATION APPROACHES 1.0. OR AS ARRIVAL RATE APPOACHES 1.0

*** THE RESULTS OF MATHEMATICAL FORMULAE GIVES A RESPONSE TIME OUTSIDE THE CONFIDENCE INTERVAL.

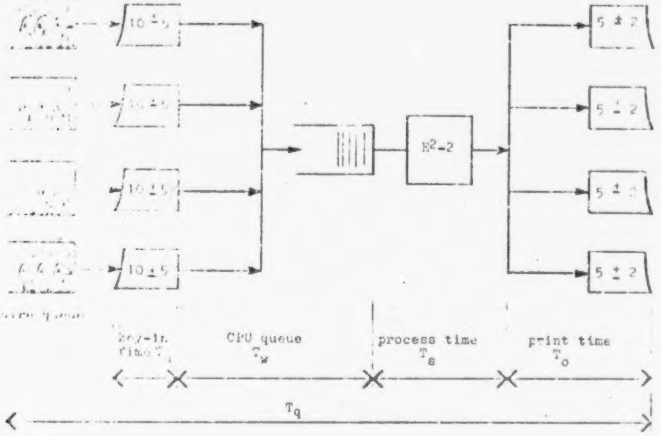
NOTE : ALL THE VALUES SHOWN ON THIS TABLE ARE THE MEAN OF ONE SIMULATION RUN OF 100 OBSERVATIONS EXCLUDING TRANSIENT OBSERVATIONS.
100 OBSERVATIONS FOR 1 ENTRY IN THE TABLE
12 ENTRIES PER CELL
9 LEVELS I.E. 8 ARRIVAL AVERAGES
3 TREATMENTS OR 3 ARRIVAL PATTERN
24 CELLS
288 ENTRIES
28.800 OBSERVATIONS

TABLE 5

```
EFFECT OF THE ARRIVAL PATTERN ON THE RESPONSE TIME IN A REAL TIME MODEL
  ANOV-/-UNWEIGHTED SOLUTION

SOURCE              SS                 DF               MS                F


     A         74915.250               7.         10702.176         191.928

  S(A)          4907.000              88.            55.761


     B           257.766               2.           128.883           1.901
    AB           495.891              14.            35.421           0.523
 BS(A)         11931.188             176.            67.791

            GRAND MEAN =        34.192

CELL MEANS--ROWS = LEVELS OF A,  COLUMNS = LEVELS OF B

                  1             2             3
      1        18.100        18.000        17.592
      2        20.050        19.633        18.733
      3        22.967        21.758        21.692
      4        28.550        28.483        25.117
      5        32.675        30.875        26.933
      6        43.775        40.025        35.617
      7        46.200        47.758        47.942
      8        70.858        66.233        69.100

NOTE:  A  =  ARRIVAL MEANS :  B  = ARRIVAL PATTERN
 A  AND  B  ARE FIXED FACTORS : A IS MAIN EFFECT AND IS CONFOUNDED WITH PLOT
```

Figure 1

A simple real-time model


10 ± 5
10 ± 5
10 ± 5
10 ± 5
E^2-2
5 ± 2
5 ± 2
5 ± 2
5 ± 2
key-in time T_i
CPU queue T_w
process time T_s
print time T_o
T_q

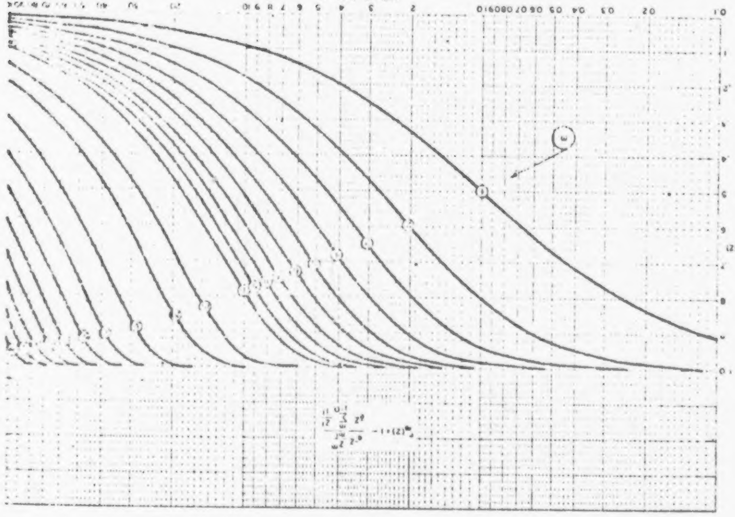


FIGURE 2

Poisson ratio function $K_n(z)$ versus service ratio z T_s T in the classical [illegible]

Source: Analysis of Some Queuing Models, (New York, IBM, Ref. F20-0007-1, 1964), p. 40.

Figure 4
Expected response time versus inquiry rate
(simulation results)

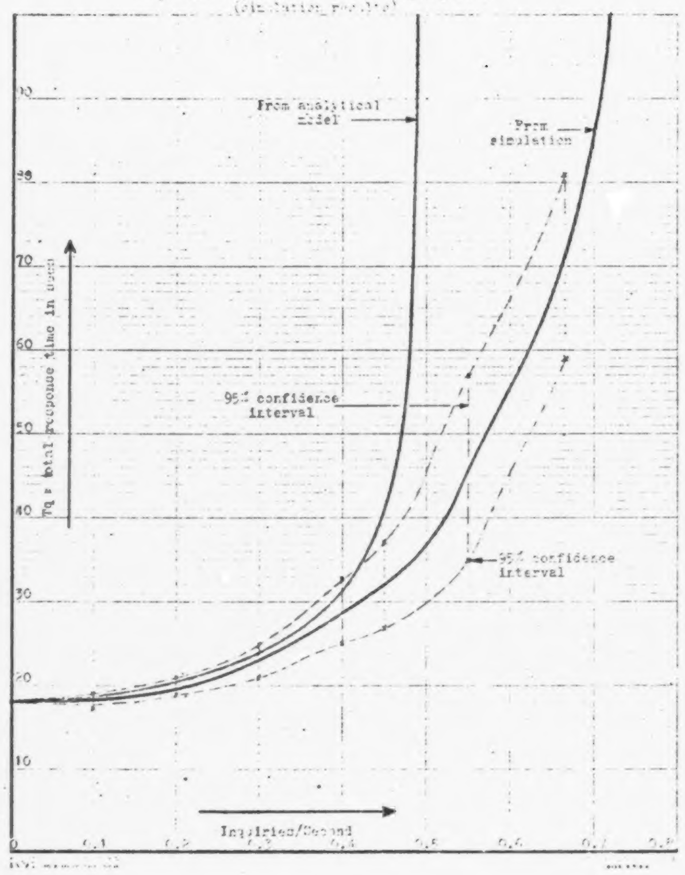
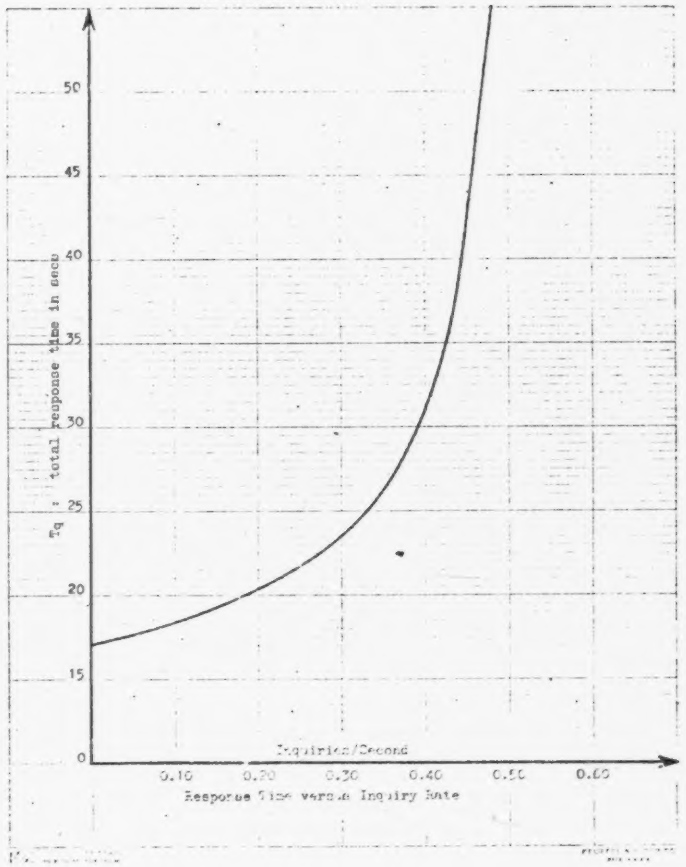


Figure 5
Expected response time versus inquiry rate
(analytical results)



MULTIPLE SEQUENCE RANDOM NUMBER GENERATORS

Joe H. Mize
Oklahoma State University

Abstract

Many discrete event simulation models utilize a single random number sequence, regardless of the number of random variables included in the model. This paper illustrates the advantages of utilizing a unique random number sequence for each random variable. An example is included to demonstrate the resulting effects. A method is described for converting a single sequence random number generator to a multiple sequence generator. An application is included that compares simulation results obtained using a single sequence of random numbers to those obtained using a unique sequence for each random variable in the model.

Introduction

Most simulation studies are concerned with systems that contain random phenomena; interarrival times, service times, demand values, time to failure, etc. In order to produce useable results, a model of such a system must provide a mechanism for representing these random phenomena.

Considerable effort has been directed toward the development of random variate generators. Most of the generators in use today make use of uniformly distributed random numbers. For examples, see Naylor, et.al. (1966), Mize and Cox (1968), and Schmidt and Taylor (1970). These random numbers, in turn, are converted to a random deviate from an appropriate probability distribution. Thus, simulation modeling requires the availability of a random number generator that will produce numbers uniformly distributed on the interval zero to unity.

The most widely used method for generating such numbers is the "congruential method," first proposed by Lehmer (1959). Almost every computer center has a random number function or subroutine on its system library. Most of these generators create a single sequence of random numbers, beginning with a user provided "seed" value.

The purpose of this paper is to demonstrate that in many simulation models a single sequence of random numbers is inadequate. This rather crucial feature of simulation modeling has not been explained sufficiently in most texts and other literature related to simulation. The two most widely used simulation languages, GPSS (see Gordon, 1969) and SIMSCRIPT (see Kiviat, et.al., 1968) provide a limited number of random number streams, however, the reasons for using multiple streams are not made clear.

Most random number generators based on the congruential method can be modified easily to produce as many parallel sequences as the user desires.

One of the prime advantages of using simulation as a means of studying complex systems is that it offers the analyst a means of comparing one system configuration to any number of other configurations. Furthermore, it is possible to construct the model such that the alternatives are evaluated using the identical sequences of event occurrences for each alternative.

Special care must be exercised in the construction of such a model, however. This is not always a straightforward process. A common pitfall of an inexperienced analyst is to believe that his simulation model compares competing alternatives against identical event occurrences, when in fact it does not. The example in the next section illustrates the nature of the problem.

Demonstrative Example

Consider the simulation of a simple queueing system in which customers arrive at random times and the length of time required for service is also random. Specifically, suppose that the time between customer arrivals is a uniformly distributed random variable over the interval 0 to 4 minutes. Service time is also uniformly distributed over the interval 1 to 6 minutes.

It is desired to simulate the performance of this system, first with one service facility and then with two. If the model is constructed properly, the performance of the system can be simulated for the two configurations such that the exact same patterns of arrivals and service times will occur. In this way, the two system configurations will be compared under identical conditions. Any differences in performance may then be attributed to the difference in system configuration (one service facility versus two) and not to "experimental error." This removes from consideration a source of variation which would be impossible to eliminate if the experiment were to be conducted in the real world.

Denote the time between customer arrivals as X, and service time as Y. It can be shown that the following process generators will produce random deviates for the two random variables, according to the probability functions specified above:

$$X = 4.0 * RN$$

$$Y = 1.0 + 5.0 * RN$$

RN is a uniformly distributed random variable on the unit interval. If our model were computerized, RN could be obtained by calling the system random number generator each time a new value of X and Y is needed.

We shall observe the performance of this model for ten arrivals to the system, first with one service facility and then with two. We will need a random number to determine the time of arrival of each customer and another to determine the service time of each customer. We will use single digit random numbers to simplify computations, although in an actual experiment we would use much greater accuracy. We will use the following sequence of random numbers:

.3, .6, .5, .5, .0, .8, .9, .7, .9,
.6, .6, .5, .9, .8, .3, .3, .4, .4,
.1, .5, .0, .5, .6, .4, .8, .9, .0,
.7, .4, .0, .9, .4, .6, .9, .7, .1,
.3, .1, .9, .9, .2, .9, .6, .2, .2,
.2, .2, .5, .0, .4

We will perform our simulation in an "event-oriented" manner. In such a simulation, process generators are used to determine when particular events will occur and all events are allowed to occur in their natural sequence.

The simulation results using a single random number sequence and one server, are shown in Table 1. The random numbers used were taken in consecutive order from the source included above. There are three numbers in each cell of the second and fourth columns of Table 1. In the upper left hand corner is simply the sequence number of the random number taken from the above source and used in the determination of the occurrence time of the indicated event. In the upper right hand corner is the random number itself. The value in the lower portion of the cell in column two indicates the time of arrival of a particular customer and is determined by substituting the random number (RN) in the upper right corner into the equation $X = 4.0 * RN$ and adding the resulting value to the arrival time of the previous customer. The value in the lower portion of the cell in column four indicates the service time for a particular customer and is obtained by substituting the random number (RN) in the upper right hand corner into the equation $Y = 1.0 + 5.0 * RN$. Column three indicates when service may begin for a particular customer and is simply the larger of the two values, Time of Arrival of that customer and End Service of the previous customer. Column five is simply the sum of columns three and four for a particular customer.

It will be instructive to trace the occurrence of a few events in the model. The arrival of the first customer is obviously the first event. The time of its occurrence is determined by using the first random number from the basic source. This gives an arrival time of 1.2. Since no previous customer is in service, the first customer may go immediately into service. We must now determine the service time for this customer by using the second random number from the basic source; $Y = 1.0 + 5.0 * 0.6 = 4.0$. Thus, customer one begins service at 1.2 and ends at 5.2.

The arrival time of customer two is determined by using the third random number in the manner described previously. It is found that customer two arrives at 3.2. This is prior to the End Service time of customer one. Thus, the service time for customer two cannot be determined at this point in the simulation. The next-event orientation of our model requires that we place customer two in a holding pattern and go on to determine the arrival time of customer three, using the fourth random number from the basic source. This results in an arrival time of 5.2 for customer three, the exact same time that customer one completes service. We now use the fifth

Table 1. Single R.N. Sequence; One Server

Customer Number	Time of Arrival	Begin Service	Service Time	End Service
0	0	0	0	0
1	1 .3 1.2	1.2	2 .6 4	5.2
2	3 .5 3.2	5.2	5 .0 0	5.2
3	4 .5 5.2	5.2	6 .8 5	10.2
4	7 .9 8.8	10.2	9 .9 5.5	15.7
5	8 .7 11.6	15.7	12 .5 3.5	19.2
6	10 .6 14.0	19.2	14 .8 5.0	24.2
7	11 .6 16.4			
8	13 .9 20.0			
9	15 .3 21.2			
10	16 .3 22.4			

Table 2. Single R.N. Sequence; Two Servers

Customer Number	Time of Arrival	Begin Service Server No.		Service Time	End Service Server No.	
		1	2		1	2
0	0	0		0	0	
1	1 .3 1.2	1.2		2 .6 4	5.2	
2	3 .5 3.2		3.2	4 .5 3.5		6.7
3	5 .0 3.2	5.2		7 .9 5.5	10.7	
4	6 .8 6.4		6.7	9 .9 5.5		12.2
5	8 .7 9.2	10.7		11 .6 4.0	14.7	
6	10 .6 11.6		12.2	13 .9 5.5		17.7
7	12 .5 13.6	14.7		15 .3 2.5	17.2	
8	14 .8 16.8	17.2		17 .4 3.0	20.2	
9	16 .3 18.0		18.0	18 .4 3.0		21.0
10	19 .1 18.4					

random number to determine the service time of customer two. Since this random number is zero, customer two departs immediately and we use the sixth random number to determine the service time of customer three. The remaining events shown in Table 1 are determined by following this same procedure.

We now wish to observe the performance of this model when two service facilities are provided. Again using the event oriented simulation procedure, the results shown in Table 2 are obtained. The first three events are the same as in Table 1. However, when customer two arrives at time 3.2, he may go immediately into service at server number two. Thus, the fourth random number in this case is used to determine the service time of customer two, whereas in Table 1 it was used to determine the arrival time of customer three. Many other discrepancies exist between Tables 1 and 2.

The critical point is that the two system configurations are being evaluated under different streams of event occurrences. For example, customer number six, who might be John Doe, arrives at time 14.0 in Table 1 and at time 11.6 in Table 2. Such a result obviates one of the principal advantages of simulation analysis.

Clearly what is required is a unique sequence of random numbers for each random variable in the model. This will permit the generation of identical streams of event occurrences, no matter how the system configuration may be modified.

Converting to Multiple Sequence Generators

It is a relatively simple matter to convert a FORTRAN random number subroutine to a multiple sequence generator. Consider, for example, the following generator:

```
      SUBROUTINE SRAND (IS, R)
      TEMP = IS
      IF (IS) 3,10,3
   3  IS = 0
      ARG = 475918104.
      IF (TEMP - 2500.) 4,4,5
   5  TEMP = TEMP - 2500.
   4  ID = TEMP
  10  IF (ARG) 11,12,11
  12  ARG = 475918104.
      GO TO 14
  11  ARG = ARG*23.
  14  IF (ARG - 1000000010.) 15,16,16
  16  ARG = ARG - 1000000010.
      GO TO 14
  15  IF (ARG - 100000001.) 17,18,18
  18  ARG = ARG - 100000001.
      GO TO 15
  17  IF (ID) 20,20,19
  19  ID = ID - 1
      GO TO 10
  20  R = ARG/100000000.
      RETURN
      END
```

As it is written, this subroutine will generate a single sequence of random numbers. The user specifies an initial burn value (an integer called IS) in the calling program. The appropriate call is as follows:

```
      CALL SRAND (IS, R)
```

When called the first time, the generator will "burn" and ignore the first IS random numbers in the sequence. Thereafter, on each call, the next number in the sequence is generated and returned as R. ARG = 475918104 is the initial seed value and 23 in statement 11 is the multiplier in the congruence relation. The generator will not allow more than 2500 values in the sequence to be burned.

The following subroutine shows the same generator, but modified to generate any number of parallel random number sequences:

```
      SUBROUTINE MRAND(IS,R,ARG,N)
      DIMENSION ARG(1)
      TEMP = IS
      IF(IS)3,10,3
 3    IS = 0
      ARG(N) = 475918104.
      IF(TEMP - 2500.)4,4,5
 5    TEMP = TEMP - 2500.
 4    ID = TEMP
10    IF(ARG(N))11,12,11
12    ARG(N) = 475918104.
      GO TO 14
11    ARG(N) = ARG(N)*23.
14    IF(ARG(N) - 1000000010.)15,16,16
16    ARG(N) = ARG(N) - 1000000010.
      GO TO 14
15    IF(ARG(N) - 100000001.)17,18,18
18    ARG(N) = ARG(N) - 100000001.
      GO TO 15
17    IF(ID)20,20,19
19    ID = ID - 1
      GO TO 10
20    R = ARG(N)/100000000.
      RETURN
      END
```

In this modified generator, the user specifies an array of burn values (called LSEED(n), all integers and each unique) in the calling program. In addition, the array ARG(n) is dimensioned in the calling program. The dimension size, n, of both LSEED and ARG is made equal to the number of unique random number streams desired. The appropriate call is as follows:

CALL MRAND (LSEED (i), R, ARG, i)

where "i" is the particular random number stream desired on this call. The initial portion of each stream is "burned" on the first call to the subroutine for that stream. On subsequent calls to that stream, the modified generator will produce the next number in the ith sequence and return it to the calling program as R. It is the user's responsibility to use the proper sequence designation as he is programming the logic of each random phenomenon.

It is noted that the modified generator presented above actually produces different portions of the same random number sequence. Suppose, for example, that we set LSEED (1) = 5 and LSEED (2) = 10. MRAND would burn the first five numbers of the sequence for random variable 1 and the first ten numbers of the same sequence for random variable 2. Thus, the 6th, 7th, 8th, etc., event occurrences for random variable 1 are governed by the same random numbers as the 1st, 2nd, 3rd, etc., event occurrences for random variable 2. This problem can be overcome by specifying in the calling program an array of multiplier values. The name of this array would be passed through the argument list and used in place of the constant 23 in statement 11 in Subroutine MRAND. In this way, a truly unique sequence is generated for each random variable. The numerical values for the multiplier array must be selected carefully, in accordance with the theory underlying the congruential method. (See Mize and Cox, 1968, or Schmidt and Taylor, 1970).

A more commonly used random number generator is that supplied in the IBM System/360 Scientific Subroutine Package. This subroutine, slightly modified, is listed below:

```
      SUBROUTINE RANDU (IX, YFL)
      IY = IX* 65539
      IF (IY) 5,6,6
5     IY = IY + 2147483647 + 1
6     YFL = IY
      YFL = YFL* .4656613E - 9
      IX = IY
      RETURN
```

IX is initialized in the calling program to any odd integer value with nine or fewer digits, and YFL is a uniformly distributed random number on the unit interval $[0,1.0]$.

This generator is easily modified to allow the generation of any number of parallel random number sequences:

```
     SUBROUTINE MRANDU (IX, YFL, N)
     DIMENSION IX(1)
     IY = IX(N)* 65539
     IF (IY) 5,6,6
 5   IY = IY + 2147483647 + 1
 6   YFL = IY
     YFL = YFL* .4656613 E - 9
     IX(N) = IY
     RETURN
     END
```

In this modified generator, the user specifies an array of seed values (called IX(n), all odd integers with nine or fewer digits and each unique) in the calling program. The dimension size, n, of IX is made equal to the number of unique random number streams desired. The appropriate call is:

CALL MRANDU (IX(i), YFL, i)

where "i" is the particular random number stream desired on this call.

Most random number generators can be modified in a similar way. Such modifications would be especially useful when using GASP or FORTRAN as the simulation language. GPSS provides the capability of generating up to eight parallel sequences, while SIMSCRIPT permits up to ten sequences.

An Application in GASP*

To illustrate the usefulness of the multiple sequence generator, the results of an application are included. The model was written in GASP (see Pritsker and Kiviat, 1969).

Two types of jobs arrive at a job shop for processing. Interarrival times and servicing times are exponentially distributed with the following means:

	Interarrival Times	Service Times
Type 1	1.25	0.50
Type 2	2.00	0.75

Type two jobs have a higher priority than type one jobs. Newly arrived type two jobs are scheduled ahead of type one jobs unless N or more type one jobs are waiting for processing, where N is an unknown number which we wish to find such that total waiting cost is minimized. Waiting costs for type one jobs and type two jobs are $1.00/minute and $3.00/minute, respectively.

Processing of type one jobs is never interrupted in order to process newly arrived type two jobs. Ten percent of type two jobs fail to pass inspection and are immediately reprocessed. Conceptually, they never move off the machine. A new service time is determined.

A GASP model was constructed for this system and run for varying values of N. Each run of the model was for 1000.0 time units.

The experiment was first conducted using a single sequence random number generator. It was then repeated using a unique random number sequence (produced by Subroutine MRAND) for each random variable in the system. The results for several values of N are plotted in Figures 1 and 2.

In Figure 1, the cost curves are very erratic and do not appear to be converging to any kind of optimum. In Figure 2, both component cost curves and the total cost curve behave very nicely, displaying

---

*The author gratefully acknowledges the assistance of Glenn C. Dunlap, Arizona State University, in the programming of this example and in the development of the multiple sequence generator, MRAND.

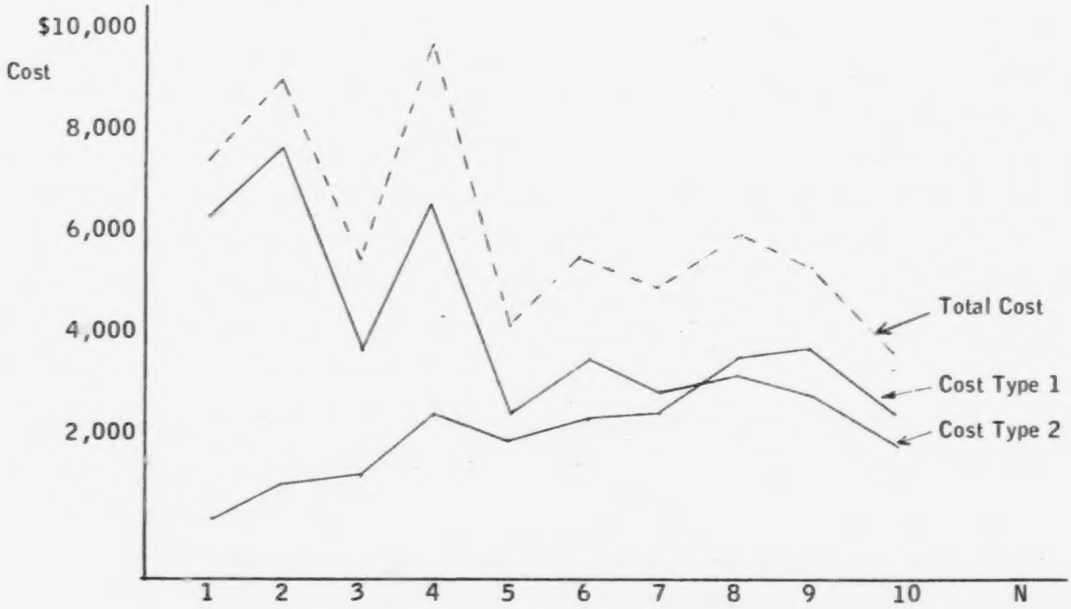


Figure 1. Cost Curves Using One R.N. Sequence

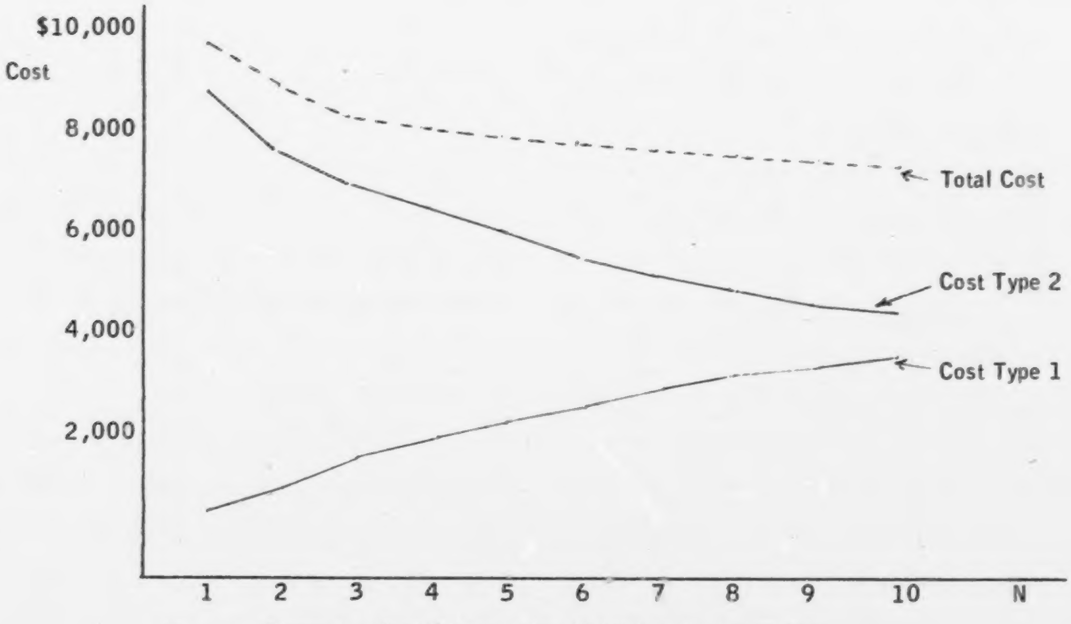


Figure 2. Cost Curves Using Multiple R.N. Sequences

the monotonic characteristics that one would expect. It is clear from Figure 2 that the total cost curve will eventually converge to an optimum value, which it does at N = 32.

Figure 3 shows the Average Number of Units in the System for several values of N. The solid line shows the results obtained using the single sequence random number generator SRAND. The dashed line shows the results obtained using the multiple sequence generator MRAND. The difference in variability is rather striking.

Summary

In many discrete event simulation models, a unique random number sequence is desirable for each random variable in the model. Failure to incorporate this feature may lead to ill-behaved models having a much larger variability than is possible with multiple sequences.

It is very easy to convert most single sequence random number generators to multiple sequence generators. The advantages of doing so are well worth the effort.

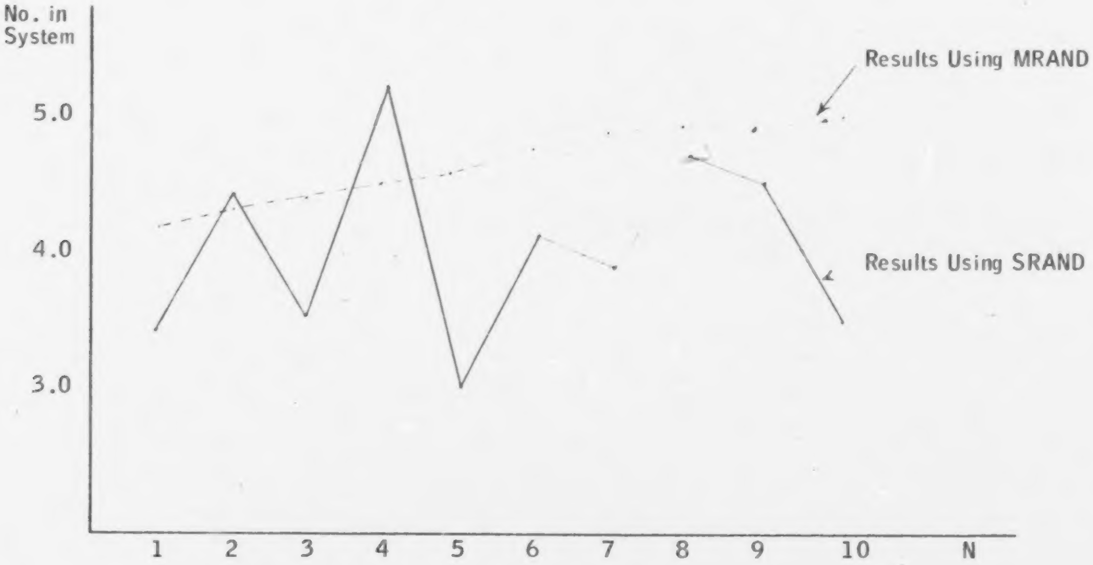


Figure 3. Average Number of Units in System for 100 Hours

References

Gordon, Geoffrey, System Simulation. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.

International Business Machine Corporation, System/360 Scientific Subroutine Package (360 A-CM - 03X) Form H20-0205-0. White Plains, New York, 1966.

Kiviat, P.J., R. Villanueva, and H. M. Markowitz, The SIMSCRIPT II Programming Language. The RAND Corporation, Santa Monica, Calif., 1968.

Lehmer, D. H., "Mathematical Method in Large-Scale Computing Units"; Proceedings of the Second Symposium on Large-Scale Digital Computing Machinery. Harvard University Press, Cambridge, Mass., 1959.

Mize, J. H., and J. G. Cox, Essentials of Simulation. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1968.

Naylor, T. H., J. L. Balintfy, D. C. Burdick, and K. Chu, Computer Simulation Techniques. John Wiley & Sons, Inc., New York, 1966.

Pritsker, A.A.B., and P. J. Kiviat, Simulation with GASP II. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969.

Schmidt, J. W., and R. E. Taylor, Simulation and Analysis of Industrial Systems. Richard D. Irwin, Inc., Homewood, Illinois, 1970.

A "WAIT UNTIL" ALGORITHM FOR GENERAL PURPOSE SIMULATION LANGUAGES

Jean G. Vaucher
Département d'informatique, Université de Montréal
Case postale 6128
Montréal, Québec, Canada

ABSTRACT

Modern simulation languages such as SIMSCRIPT II and SIMULA 67 are very powerful general purpose languages which contain facilities to handle lists and to schedule events in simulated system time (imperative sequencing statements). These languages do not include some of the useful but more specialized features of previous languages (GPSS, CSL, SOL) especially interrogative sequencing statements such as "SEIZE <facility>" or "WAIT UNTIL <Boolean expression>"; however, the definition capability of the new languages is powerful enough to permit their extension to include the interrogative features.

The addition of some features of GPSS to SIMULA 67 was presented at a previous SIMULATION CONFERENCE. The present paper extends that work by describing an efficient algorithm which adds the "WAIT UNTIL" procedure to SIMULA.

INTRODUCTION

Modern simulation languages such as SIMSCRIPT II [1] and SIMULA 67 [2] are very powerful general purpose languages which contain a relatively small number of special features required for simulation. In common with other modern languages, they have the ability to define complex data structures, allocate memory dynamically and handle lists. The special features are mainly concerned with simulated system time. Each language provides a CLOCK and maintains a list of event notices in chronological order. Each also contains imperative sequencing statements of the form

"SCHEDULE AN event AT timex".

The compilers generate fairly efficient code and the languages can therefore be used for general purpose computing as well as simulation. The power and flexibility of the languages are such that they may be used for any type of simulation.

When comparing the two selected modern languages to some older languages such as GPSS, CSL, SOL etc..., it can be seen that the gain in generality and efficiency has not been attained without some losses. The predefined objects of SOL and GPSS such as facilities, transactions and storages, are not present and interrogative scheduling statements have also disappeared. Interrogative statements are used when it is impossible to predict in advance the system time when an event should take place. An example of such a statement commonly used in GPSS is "SEIZE". If a transaction attempts to SEIZE an occupied facility, its execution is halted until the facility becomes free and it is not possible to know in advance when this event will take place.

A much more general interrogative scheduling statement is found in SOL [3]. This statement has the form

"WAIT UNTIL <boolean expression>"

where the boolean expression is a condition which must be met before execution continues. The condition may refer to any number of state variables.

For certain classes of problems the availability of predefined objects and interrogative statements allows models to be described in a very natural and concise fashion. Programs for the same models written in SIMULA 67 or SIMSCRIPT II will be much longer, complex and error prone although they will tend to use less computer time and memory space.

It might appear that the modern languages are hardly better than FORTRAN, perhaps augmented with some subroutines (GASP), when it comes to simulation. Such hasty judgement overlooks the definition capability of the new languages where it is possible, using the source language, to define the data structures and procedures required to implement the useful features that were frozen in the older languages. Given a sufficiently powerful general purpose language, these extra features can be added to the language so neatly that they appear to be extensions to the compiler.

SIMULA 67 is particularly well suited to this extension philosophy. It contains several elegant mechanisms whereby precompiled routines and object declarations can be added to a user programme. This has already been done at the University of Montreal, adding to SIMULA useful objects of GPSS, such as transaction and facilities, as well as the associated procedures such as SEIZE and RELEASE [4]. The present paper describes the further addition of a WAIT UNTIL feature to SIMULA. The algorithm used to implement this feature is easily described in SIMULA. The simplicity of the description shows quite clearly the sequencing problems inherent in such a powerful feature and permits experimentation with alternative algorithms.

The paper first gives a brief description of some pertinent features of SIMULA 67, then shows some examples of the use of WAIT UNTIL. The implementation is described and the problems arising from the use of TIME as part of the WAIT CONDITION are pointed out. The problems are partially resolved through the use of ALARMS. Finally, there is a discussion of efficiency considerations.

SIMULA 67

SIMULA 67 was developped by Dahl & Nygaard as a successor to ALGOL. In appearance, the language is much like ALGOL although some of the weaker points of its ancestor have been redesigned and other features have been added.

SIMULA 67 has list handling capability and standard list procedures FIRST, SUC and EMPTY will be used in this paper. The genitive or dot notation is used in SIMULA to indicate to which object a procedure should be applied. For example,

"LIST 1.FIRST"

means the FIRST element of LIST 1.

One of the more important lists in the system, the list of event notices or sequencing set (SQS) is maintained automatically by the system and simulated system time is given through the procedure TIME.

One of the important features of SIMULA is the use of PROCESS as a data type. Processes are based on the co-routine concept. Each process has a local instruction counter in addition to its local data and programme. Processes operate in quasi-parallel fashion in roughly the same way as programmes in a multi-programming environment.

Execution of a process may be controlled from another process by scheduling statements such as the following:

1) ACTIVATE process AT timex ;
2) ACTIVATE process DELAY dt ;
3) ACTIVATE process ;

In the first two statements, an active phase for the "process" is scheduled at some time in the future. The last statement is almost equivalent to a procedure call in that "process" is activated immediately, interrupting execution of the activating process; this is called "direct" scheduling.

A process can also schedule itself with the procedures HOLD and PASSIVATE. "HOLD (ts)" causes the process to halt for an interval "ts". PASSIVATE suspends execution of the process for an indefinite period; in this case, the process continues execution only when activated by another process.

Processes can be placed into or removed from lists with the procedures INTO (list) and OUT.

Figure 1 gives a simple but complete SIMULA program using some of the features described. The main item of interest is the definition of CLIENT which is a process. Clients arrive at intervals of 10 minutes, spend a time TS in the system and leave. TS could be any of SIMULA's random generator procedures. There is no contention for resources so that no queues form. N represents the number of customers in the system. The main program activates the first client and, thereafter, each client activates his successor. The main program halts for 1000 minutes giving a duration of 1000 minutes to the simulation.

```
SIMULATION  begin
   process class client ;
   begin
      activate new client delay 10 ;
      N := N + 1 ;
      hold (TS) ;
      N := N - 1 ;
   end ;
   integer N ;
   activate new client delay 0 ;
   hold (1000) ;
end ;
```

FIGURE 1 - A complete SIMULA program

It is interesting to note that list handling facilities as well as simulated system time are not strictly part of the basic SIMULA language. These facilities are defined as a standard precompiled extension to the language called SIMULATION. A user indicates to the compiler that he wishes to use this extension by prefixing his program with the name of this extension: hence, the "SIMULATION begin" at the start of the program. In the same way, the standard entities of GPSS and the WAIT UNTIL procedure have been implemented as an extension called GPSSS. To make use of these extra facilities, a user would prefix his program with the name of the extension, "GPSSS".

USE OF "WAIT UNTIL"

This section presents the "WAIT UNTIL" statement as a natural way of expressing complex scheduling rules. Three examples of increasing complexity will be used.

The WAIT UNTIL statement is implemented as a procedure with one parameter which can be termed the "wait condition":

WAIT UNTIL (condition) ;

The condition is a Boolean expression of any degree of complexity. It may even include function calls with side effects. When the procedure is used by a process, further execution is suspended until the condition becomes true. If the condition is true at the outset, there is no wait.

The first example is a modification of the programme of Figure 1. In this example, clients need the use of a facility whose state, free or occupied, is represented by the boolean variable "free 1". Clients wait until, the facility is free then mark it as busy by setting "Free 1" equal to false. They use the facility for a time TS then reset the facility to free upon leaving the system. Here the statement, WAIT UNTIL (free 1) is equivalent to a SEIZE of GPSS. The PROCESS definition for this example is shown in Figure 2.

```
process class client ;
begin
    activate new client delay 10 ;
    N := N + 1 ;
    WAIT UNTIL (free 1) ;
    free 1 := false ;
    hold (TS) ;
    free 1 := true ;
    N := N - 1
end ;
```

FIGURE 2 - Use of WAIT UNTIL as SEIZE

In the next example shown in Figure 3, a client must use two facilities one after the other. The facilities are represented by "free 1" and "free 2" and the order of use is immaterial.

```
process class client ;
begin
        ---
        ---
      WAIT UNTIL (free 1 or free 2) ;
      if  free 1 then goto one then two
                 else goto two then one ;
one then two:
      free 1 := false ;
      hold (TS1) ;
      free 1 :=  true ;
      WAIT UNTIL (free 2) ;
      free 2 :=  false ;
      hold (TS2)
      free 2 :=  true ;
      goto to over ;
  two then one :
         - - -
         - - -
```

FIGURE 3 - Scheduling dependent on two facilities

Although the use of system time, TIME, as part of the "wait condition" will be shown to present problems, it is often useful to model "balking" where a customer will leave a queue if he has not been served within a certain time.

In the last example, the client wants to use facility I but refuses to wait in line longer than TWAIT unless he sees that he will be the next to be served. In this case, the facility is represented by two integer variables, IN1 and ØUT1, as well as by the state indicator, "free 1". These variables contain the total number of clients having arrived at the facility and having left the facility. The difference between them indicates the number of clients either being served or waiting for service. The client notes in AHEAD the value of IN1 at the time of arrival; then AHEAD-ØUTI will give the number of people ahead in the queue.

```
process class client ;
begin
     integer AHEAD ;
     real TØUT ;
       ---
       ---
     AHEAD := IN1 ;
     TØUT := TIME + TWAIT ;
     WAIT UNTIL (free 1 or
          (TIME > TØUT AND AHEAD - ØUT1 >1));
     ØUT1 := ØUT1 + 1 ;
     if free 1 then goto occupy 1
               else goto exit ;
       - - -
       - - -
end client ;
```

FIGURE 4-Programming of complex balking decision

The WAIT UNTIL statement is clearly a powerful tool for the description of models. It also describes scheduling in a natural manner.

IMPLEMENTATION

The WAIT UNTIL extension has been implemented using the source language facilities of SIMULA and not by modifying the compiler. There are four vital elements to the implementation of WAIT UNTIL:

1) A procedure called WAIT UNTIL to be used by the processes.
2) A list, WAITQ, where processes halted as a result of use of the WAIT UNTIL procedure are kept.
3) a monitor which examines waiting processes and reactivates them at the opportune moment.
4) A global Boolean variable, ACTION, used for communication between waiting processes and the monitor.

The procedure makes use of parameter passing by name. In the procedure, each use of "B" results in re-evaluation of the "wait condition" passed as the parameter. The procedure is given in Figure 5.

```
1  procedure WAIT UNTIL (B) ; name B ;
   boolean B ;
2  begin
3       if B then goto exit ;
4       into (WAITQ) ;
5       if monitor.idle then activate
        monitor after next ev ;
6  loop : passivate ;
7       if not B then goto loop ;
8       out ;
9       action := true ;
10 exit :
11 end wait until ;
```

FIGURE 5 - The WAIT UNTIL procedure

If the wait condition is true initially, the process is not halted and leaves the procedure. Otherwise, the process is placed in WAITQ and passivated. The test to determine if a process may leave WAITQ is done, not by the monitor but by the process (lines 6,7). However, it is the responsibility of the monitor to activate processes when it is possible that the wait condition may be true. When the wait condition is fulfilled, the process leaves WAITQ by using the standard SIMULA procedure OUT (line 8), sets "action" (line 9) to indicate successful exit and carries on execution of its programme. To reduce overhead, the wait monitor is normally idle (passive) and is only activated when processes enter WAITQ (line 5).

The interrogation of the wait conditions could be done continuously, but this is grossly inefficient. Once a process is blocked, it can only continue following a change in system state. With the exception of the system variable TIME, a change of system state may only be caused by an event. Processes in WAITQ need, therefore, only be queried after each event. This periodic examination is controlled by the "wait monitor" whose description is given below:

```
1   process class wait monitor ;
2   begin
3        ref (process) pt ;
4   start : if waitq.empty then passivate ;
5        action := false ;
6        pt :- waitq.first ;
7   loop : if pt== none then goto wait ;
8            activate pt ;
9            if action then goto start ;
10           pt :- pt.suc :
11           goto loop ;
12  wait : reactivate this wait monitor after
             next ev ;
13           goto start ;
14  end      wait monitor ;
```

FIGURE 6 - The wait monitor

Basically, the monitor goes into action

after each event (line 12) and activates in turn all processes in WAITQ (lines 7-11). The implicit priority is FIFO, new arrivals being placed at the end of WAITQ but a priority scheme could easily be implemented.

If no process is able to leave, the monitor waits until the next event. If, on the other hand, a process leaves WAITQ, this is equivalent to a new event and the monitor passes through WAITQ once again. The monitor carries on testing until no process can advance. It is by testing the boolean "action" that the monitor checks if a process has been able to leave WAITQ.

To reduce overhead, the monitor passivates itself when it finds WAITQ empty. It is reactivated by the first process to enter WAITQ.

PROBLEM OF TIME

The implementation described is very general and works for any possible wait condition with one exception. The exception is the use of the variable TIME. TIME is different from all other system variables in that it changes continuously by itself without posting event notices. This causes some difficulties. Consider the two following statements:

a) WAIT UNTIL (TIME = 15) ;

b) WAIT UNTIL (TIME > 15) ;

and assume that two future events have been scheduled at times 10 and 20 respectively. The processes in WAITQ will only be examined at these two times. The process having executes statement (a) will never be reactivated and the process of statement (b) will carry on at time = 20. This is obviously contrary to the intent of the programmers who intended for their processes to carry on when TIME became equal to 15.

One solution would be to scan WAITQ at fixed time intervals Δt and require that wait conditions involving TIME expressions work with multiples of Δt. This method throws away the advantages gained by using an event-oriented simulation.

The solution that has been adopted is to provide the programmer with dummy events called ALARMS. These are defined in SIMULA by

process class ALARM ; ;

A programmer can now insure correct operation of the "balking" example of Figure 4 by generating an ALARM to trigger an event when the client may wish to leave. The proper procedure is shown below:

```
TOUT := TIME + TWAIT ;
activate new ALARM at TOUT ;
WAIT UNTIL (free 1 or TIME = TOUT) ;
```

DISCUSSION OF EFFICIENCY

Interrogative scheduling as examplified by the WAIT UNTIL statement is a powerful tool, but in the case of simple scheduling decisions, the method is highly inefficient compared to other scheduling algorithms. The inefficiency comes from two main sources.

1) Waiting transactions are placed in a single list irrespective of the wait condition. A change of state in the system leads to examination of all waiting processes.

2) Each waiting process must be tested before knowing if it can continue. A large proportion of the tests will be unsuccessful.

Efficiency is gained by reducing the number of waiting processes that must be examined for a given change in system state. This is achieved by having several wait lists, each corresponding to a distinct wait condition (this is equivalent to the inverted list concept used in structuring data bases). A natural ordering within the list so that only the first or last element need be tested also reduces overhead. If the wait condition for a particular list is specific enough, the repetitive individual testing of each waiting process may be eliminated.

The future events list and imperative sequencing statements found in most simulation languages form an example of these principles. The

The list contains all the processes waiting for the passage of time. Control may be passed to the first element in this list without searching or testing. It is therefore much more efficient to use

"activate this process at T0UT;"

than to use

"wait until (TIME = T0UT);"

When scheduling of a process depends on the state of one variable only, efficiency may be gained if the variable is programmed as a GPSS facility or storage. Efficient implementation in SIMULA of these entities as well as the associated SEIZE and RELEASE procedures has previously been described [4]. It would then be advisable to use

"SEIZE (1);"

rather than

"wait until (free 1);"

It is only in the case of complex scheduling decisions such as presented in the "balking" example that WAIT UNTIL should be used. In such cases, putting a waiting process in several lists, each corresponding to a variable involved in the waiting condition leads to intolerable administrative overhead. If function calls are allowed as part of the wait condition, it may even be difficult to find all the variables involved.

in these cases, the WAIT UNTIL algorithm presented is a useful compromise. It centralizes the scheduling process and makes it systematic. "Ad hoc" methods to give equivalent results are certainly more prone to error and do not appear likely to result in greater efficiency.

CONCLUSIONS

The WAIT UNTIL statement has been shown to be a powerful scheduling tool. It also leads to a natural description of many situations. This feature, present in older languages, is absent from the modern generation of general purpose simulation languages.

WAIT UNTIL has been implemented as a source language extension to SIMULA 67. It could similarly be added to other languages, although the implementation might not be as elegant. Efforts were made to minimize the overhead resulting from use of this feature. The WAIT UNTIL statement is more suitable to complex decision rules. Simple scheduling can be accomplished more efficiently in other ways.

The use of the variable TIME as part of the wait condition was shown to give rise to some problems which could be eliminated by means of dummy event called ALARMS.

ACKNOLEDGEMENTS

This work was supported by the National Research Council of Canada.

REFERENCES

[1] Kiviat P.J., Villanueva R., Markowitz H.M., "The Simscript II programming language", Prentice Hall (1968).

[2] Dahl O.J., Myhrang B., Nygaard K., "SIMULA 67 common base language", Publication S22, Norwegian Computing Centre, Forskningsveien 1B, Oslo 3 Norway.

[3] Dahl O.J., "Discrete event simulation languages" pp. 349-395 in "Programming Languages", F. Genuys (Editor), Academic Press, London (1968).

[4] Vaucher J.G. "Simulation data structures using SIMULA 67", pp. 255-260, 1971 Winter Simulation Conference, New York, Spensored by ACM, AIIE, IEEE, SHARE, SCI, TIMS.

Session 3: Inventory and Distribution Models
Chairman: Ernest Koenigsbert, University of California

Inventory control is one of the "classical" problem areas for quantitative analysis. Inventory and distribution systems have been subjected to analysis using simulation since the advent of medium size computers. In this session the emphasis is on production inventories and their interaction with production efficiency and warehouse operations.

Papers
"Simulation of Sequential Production Systems with In-Process Inventory"
David R. Anderson, Brian D. Sellers,
Mohammed M. Shamma, University of Cincinnati

"A Model for Analyzing Closed-Loop Conveyor Systems
with Multiple Work Stations"
Lynn E. Bussey, M. Palmer Terrell,
Kansas State University and Oklahoma State University

"A Cased Good Conveyor Simulator"
Donald A. Heimburger, The Procter and Gamble Company

"A Simulator for Designing High-Rise Warehouse Systems"
Kailash M. Bafna, Georgia Institute of Technology

Discussant: Roland Young, Del Monte Corporation

SIMULATION OF SEQUENTIAL PRODUCTION SYSTEMS WITH IN-PROCESS INVENTORY

David R. Anderson
Brian D. Sellers
Mohammed M. Shamma
Department of Quantitative Analysis
University of Cincinnati

Abstract

This paper presents simulation results from a general sequential production system. The results are used to establish the effect of service time variability and to estimate minimum cost-in process inventory capacities.

This paper deals with the problem of finding optimal in-process inventory levels for a general production system. Specifically the system can be described as a production line with N separate stages (work stations) where an in-process inventory buffer with a fixed capacity is provided between these stages. All work units are processed through the stages in a fixed sequence. We assume an infinite supply of input at the first stage and no blocking of output from the last stage. Typically, such systems are used for high volume production, and operating costs saved by choosing an optimal size buffer will be desirable. Industrial engineers and system analysts are frequently confronted with the design and evaluation of such production line systems.

The Model

In our research we simulated 2-, 3-, and 4-stage production systems with 0, 2, 4, 6, and 8 buffer capacities and with normal service times. Coefficients of variation for the normal distributions ranged from .01 to .30. We used the normal distribution for several reasons: numerous variables seem to follow a pattern of variation that is similar to the normal distribution; and the normal distribution can be an excellent approximation to several other distributions. The chief reason for using the normal distribution, however, was its practical significance. Lind [6] and Nadler [7] found that manufacturing processes, whether machine or operator controlled, exhibit an inherent variation about their mean production rates ranging from approximately normal distributions to positively skewed distributions of the Pearson Type II curve. GPSS (General Purpose Systems Simulation) was chosen as the language for our studies because of its adaptability to manufacturing processes and especially to production lines. Not only is the structure of GPSS suited quite well to such queueing problems incapable of mathematical formulation, but it also permits the direct and complete observation of the dynamic behavior of the processes. In general simulation provided a closer fit to reality and an insight into system characteristics unobtainable through strictly analytical formulations.

In this research we investigated the behavior of systems in which the individual stages have service times with different coefficients of variation. For each system we determine an overall measure of system variation. We define the system coefficient of variation by the following formula:

$$\overline{CV} = \sqrt{\frac{(CV1)^2 + (CV2)^2 + \ldots + (CVN)^2}{N}} \qquad [1]$$

where $\overline{CV}$ = the system coefficient of variation

CV1 = coefficient of variation of Stage 1

CV2 = coefficient of variation of Stage 2

.
.
.

CVN = coefficient of variation of Stage N

N = number of stages in the system

The service time at each stage in the simulated production line was randomly assigned a normal distribution with a coefficient of variation of .01, .02, .03, .04, ..., .27, .28, .29, or .30. Three sets of variation patterns for the coefficient of variation of each system were calculated, and one of them was discarded on the basis of duplication or close similarity to another pattern. The example below illustrates two such patterns for a 3-stage model with a buffer capacity of 4 and a coefficient of variation for the system of 0.20.

	CV Stage 1	CV Stage 2	CV Stage 3	$\overline{CV}$ for System
System 1	.20	.05	.28	.2007
System 2	.24	.24	.08	.2013

The Simulation Process

A basic unit called a transaction travels through the simulation model with processing stations and storage areas, and statistics are gathered on its movement with respect to congestion and occupancy, total time, and delay. We used a 3000 transaction starting run for assurance that the effect of transient-state build up would not affect the steady-state statistics. We then used 15,000 transaction steady-state runs.

Two types of delay were identified in the studies: lack of work and blocking delay. Lack of work occurs at a stage when the buffer immediately preceding it is empty and the stage is available for processing; blocking occurs when the buffer immediately proceeding it is full and the stage has completed processing on its current contents. During the simulation runs we gathered statistics on blocking and lack of work delay, buffer content, facility utilization, and production times. After gathering statistics for all the systems, we used regression analysis for the formulation of functions defining various aspects of the systems according to the parameters of the number of stages in the system, the system coefficient of variation, and the buffer capacity. Our goal was to use the simulation results to find the following functional relationships:

$$\begin{aligned}\text{INVENTORY} &= f\,(N,B,\overline{CV}) \\ \text{DELAY} &= f\,(N,B,\overline{CV}) \qquad [2] \\ \text{UTILIZATION} &= f\,(N,B,\overline{CV})\end{aligned}$$

where N = number of stages in system

B = buffer capacity

$\overline{CV}$ = system coefficient of variation

Using these equations, we developed the general cost equation for any production line system where

$$\text{TOTAL COST} = f\,(N,B,\overline{CV}) \qquad [3]$$

This equation along with the others provided the framework for determining optimal buffer capacities, optimal operating costs, and optimal utilization of production line facilities.

General System Behavior

In viewing the data, it appears that buffer capacity and the system coefficient of variation have the most significant effect on average delay. As buffer capacity increases, delay rapidly approaches zero. There is a very large drop in delay with an increase in buffer capacity from zero to two units. This is in agreement with Hatcher's analytical results (1969) that only a small amount of storage capacity is required to reach near optimum production rates. For example, a 3-stage system with a system coefficient of variation of .20 displayed 13.3% delay for no buffer capacity, 2.11% for a buffer capacity of two, and .73% for a buffer capacity of eight. Similarly as the system coefficient of variation increases, the system displays greater delay. A 3-stage system with a buffer capacity of 4 displayed .12% average delay for a system coefficient of variation of .05 and 1.9% delay for a coefficient of .25. Increasing the number of stages in the system also increased delay but not as significantly as the other two variables. For example, for two systems with buffer capacities of 4 and a system coefficient of variation of .20, a 2-stage system displayed 1.0% delay and a 4-stage system displayed 1.7% delay.

Average system content appears to be affected by both buffer capacity and number of stages in the system. Naturally system content increases as buffer capacity is increased up to a point where blocking delay approaches zero. When buffers are large and blocking delays near zero, further increases in buffer capacity remain unused. Obviously an increase in the number of stages increases average system content by more than one unit since not only is another stage added but also another buffer. In viewing the data, however, the system coefficient of variation appears to have no significant effect on average content. For the case of no buffer, systems with higher system variation display slightly lower average content, but this does not hold as soon as buffer capacity is added.

In analyzing internal system behavior, the results for lack of work and blocking delay were in agreement with Anderson's earlier results where the coefficient of variation was held constant at each stage throughout the system. Basically, blocking delay is highest for the first stage and decreases at each stage up to the last where it is zero. Lack of work is zero for the first stage and increases up to the highest amount at the last stage. This general rule held for all systems with few exceptions. Average buffer content was, with only one exception, highest in the first buffer and lowest in the last buffer. The percentage of time the buffers were empty, a good indication of relative lack of work delay at proceedings stages, was in every case lowest for the first buffer and highest for the last buffer. In some cases the gradations from lowest lack of work and blocking delay were gradual and in other cases quite steep. However, this could not be traced to particular variation patterns. Also, evidence seems to indicate that contents of the individual buffers is independent of the system coefficient of variation.

Estimating System Delays and Contents

First we used stepwise regression to develop a formula for content. Using the following terminology

C = system coefficient of variation
N = number of stages
B = buffer capacity

we regressed the variables C, N, B, C^2, N^2, B^2, CN, CB, and NB. We obtained a correlation coefficient of .938. With the exception of B^2, all of the variables containing C were the last to enter the regression equation and did not increase the correlation coefficient significantly. Using the variables N, B, and NB, we obtained the following equation:

$$\text{Content} = .08-.27B+.93N+.41NB \qquad [4]$$

with an R^2 of .928. Using nonlinear regression did not result in any significant improvement.

After this we developed a delay equation from stepwise regression of the variables B, N, C, B^2, C^2, N^2, BN, BC, and NC. But we obtained an R^2 of only .881 and the fit was very poor. Next we resorted to nonlinear regression. As a theoretical basis for the delay equation, we considered Hunt's analytical derivation of delay in the 2-stage exponential service time system. He obtained analytically the following equation:

$$\text{Delay} = \frac{1}{B+3} \qquad [5]$$

Using this starting point, Anderson had previously developed the delay equation

$$\text{Delay} = \frac{a_1}{B+a_2} \qquad [6]$$

where $a_1 = f_1(B,N,C)$
$a_2 = f_2(B,N,C)$

After testing several formulations, we came up with the following function which is relatively simple

$$\text{Delay} = \frac{1}{B+.453}\,(-.134 +.131N^{.028}+.111C^{.870}+.052CN) \qquad [7]$$

with an R^2 of .985. Further complexity did not improve the fit significantly.

The Cost Model and Optimal Inventory Level

In order to evaluate systems on a cost basis and derive optimal buffer capacity for various cost structures, we use the same cost model as presented in Anderson's earlier paper. The general model for a sequential queue is shown below.

Let N = number of stages

B = buffer capacities

D = average delay/unit time

I = average contents of the system

S = total number of storage spaces

K = cost of delay/unit time

L_1 = inventory cost/unit/unit time

L_2 = storage space cost/unit/unit time

T = total cost/unit time

the

$$T = DK + IL_1 + SL_2 \qquad [8]$$

But since S is known to be:

$$S = (N-1)B \qquad [9]$$

we have

$$T = DK + IL_1 + (N-1)BL_2 \qquad [10]$$

Using the equations we obtained from regression analysis for average delay and content, we can formulate the total cost equation where a_1 and a_2 are as defined before

$$T = \frac{a_1\ K}{B+a_2} +(.08-.27B+.93N +.41NB)L_1+(N-1)BL_2 \qquad [11]$$

For optimal buffer capacity with respect to cost we have

$$\frac{\partial T}{\partial B} = \frac{-a_1\ K}{(B+a_2)^2} -.27L_1+.41NL_1 +(N-1)L_2 = 0 \qquad [12]$$

$$B^* = \sqrt{\frac{a_1\ K}{-.27L_1+.41NL_1+(N-1)L_2}} -a_2 \qquad [13]$$

where $a_1 = -.134+.131N^{.028}+.111C^{.870} +.052CN$

$a_2 = .453$

The second derivative shows this to be a minimum.

In order to compare the cost properties of the two models, let L equal the effective space-holding cost at the optimal buffer size of B*. From the equation we must have

$$-.27L+.4NL+(N-1)L=.27L_1+.41NL_1 +(N-1)L_2 \qquad [14]$$

giving

$$L = \frac{-.27L_1+.41NL_1+(N-1)L_2}{1.41N -1.27} \qquad [15]$$

Rewriting the equation, we get

$$B^* = \sqrt{\frac{a_1\ K}{(1.41N-1.27)L}} - a_2 \qquad [16]$$

Let ϕ = the ratio of delay cost K to the effective holding cost

$$\phi = K/L \qquad [17]$$

Then

$$B^* = \sqrt{\frac{a_1\ \phi}{(1.41N-1.27)}} - a_2 \qquad [18]$$

and

$$\begin{aligned} T^* &= DK + LB + (N-1)B^*\ L \\ &= DK + IK/\phi + (N-1)B^*\ K/\phi \\ &= (D + I/\phi + (N-1)B^*/\phi)\ K \qquad [19] \end{aligned}$$

Letting $K=\phi$ we generate the following table for optimal buffer capacities and optimal costs for the given ratios of ϕ.

		Buffer Sizes					Total Cost				
		$\overline{CV}$					$\overline{CV}$				
ϕ	N	.05	.10	.15	.20	.25	.05	.10	.15	.20	.25
1,000	2	2	4	4	5	6	10.32	13.74	16.33	18.44	20.35
	3	2	3	3	4	5	15.75	20.92	25.01	28.11	31.12
	4	2	2	3	4	4	21.01	28.01	32.75	37.21	40.91
5,000	2	6	9	10	12	13	21.28	29.08	34.89	39.71	43.94
	3	5	7	8	10	11	33.34	44.83	53.64	61.01	67.44
	4	4	6	7	9	10	44.61	59.43	71.08	80.76	89.31
10,000	2	9	12	15	17	19	29.60	40.57	48.57	55.64	61.62
	3	7	10	12	14	15	46.51	62.76	75.17	85.59	94.72
	4	6	9	11	12	14	62.22	83.28	99.62	113.38	125.46
50,000	2	20	28	34	39	43	64.62	89.17	107.57	122.88	136.25
	3	17	23	27	31	35	102.14	138.46	166.20	189.47	209.88
	4	15	20	25	28	31	136.58	183.99	220.49	251.19	278.23

Simulation Summary

In total we simulated the production process of 18,000 units a total of 150 times. These simulations required over seven hours of computer time and cost roughly $4800.

Conclusion

A general class of simulation models of production lines have been studied to formulate theories on system behavior. All of the systems had normally distributed processing times as an approximation to conditions often encountered in the manufacturing environment. Results show that the behavior of such systems is a function of buffer capacity, the system coefficient of variation, and the number of stages in the system, and that the pattern of variation among the individual stages has no significant effect on the behavior of the system. From the results, it appears possible to obtain an excellent approximation to optimum buffer size for any system meeting the general assumptions of the model. Also we can gain insight into the cost of constraints on buffer sizes less than optimum. Simulation has provided significant results to the general in-process inventory problem where a theoretical approach would have been virtually impossible.

References

1. Anderson, David R., "Simulation Studies on Sequential Queues", Proceedings American Institute of Decision Sciences Southeastern Conference, July, 1971.
2. Barten, Kenneth, "A Queueing Simulator for Determining Optimum Inventory Levels in a Sequential Process", Journal of Industrial Engineering, Vol. VIII, No. 4, July-August, 1962.
3. Buchan, Joseph and Koenigsberg, Ernest, Scientific Inventory Management, Chapter 22, Prentice-Hall, 1963.
4. Hatcher, Jerome M., "The Effect of Internal Storage on the Production Rate of a Series of Stages Having Exponential Service Times", A.I.I.E. Transactions, Vol. I, June, 1969.
5. Hunt, Gordon C., "Sequential Arrays of Waiting Series", Operations Research, December, 1956.
6. Lind, Warren E., "A Statistical Analysis of Work-Time Distributions", MSIE Thesis, Georgia Institute of Technology, June, 1953.
7. Nadler, Gerald, Motion and Time Study, Chapter 20, New York, McGraw-Hill Book Company, 1955.
8. Young, H. H., "Optimization Models for Production Lines", Journal of Industrial Engineering, January, 1967.

A MODEL FOR ANALYZING CLOSED-LOOP CONVEYOR SYSTEMS WITH MULTIPLE WORK STATIONS

Lynn E. Bussey, Ph.D.
Kansas State University
Manhattan, Kansas

M. Palmer Terrell, Ph.D.
Oklahoma State University
Stillwater, Oklahoma

Abstract

This paper presents the modeling methodology incorporated in a GPSS/360 program to simulate and test the operation of a generalized recirculating conveyor-supplied system, consisting of loader, conveyor, and multiple work stations. Features of the generalized model include its ability to simulate a constant-speed closed-loop conveyor with discretely spaced random loads, variable spacing between work stations, variable supply and return distances, variable total length, zero or specifiably finite local storages at each work station, homogeneous service rates in the work stations, variable number of work stations, and choice of arrival (loading) distribution. Output statistics include many tabulated operating characteristics of the system.

Introduction

The design of recirculating conveyor-supplied multiserver systems constitutes a significant area of responsibility for engineers in manufacturing organizations. Since investments in conveyor systems are expensive, it is desirable that engineers have a reliable design methodology for such systems.

A search of pertinent conveyor theory literature indicates that a number of prior investigators apply multichannel queueing theory or theory-based simulations to the analysis of recirculating conveyor-supplied multiserver systems. Previous research efforts on the conveyor system typically have investigated one of two situations: (1) conveyors with randomly-spaced loads serving single or multiple servers (e.g., a powered belt conveyor), and (2) conveyors with discretely spaced loads (usually uniformly spaced) serving single or multiple servers (e.g., a powered line or overhead-type conveyor with "hooks").

The methods used to analyze the discretely spaced ("hook") conveyor are completely different from those used for the randomly spaced ("belt") type. The present model is concerned with the analysis of the recirculating "hook"-type conveyor, in which loads can be visualized as being spaced at integer multiples of the "hook" spacing on a closed-loop conveyor that moves at constant speed. While there is a fair amount of prior research concerned with non-recirculating conveyor-supplied systems, there appears to be only a meager amount of work concerned with the recirculating system in which the conveyor is a continuous loop moving at constant speed.

Kwo (2), 1958, was the first to conceive the recirculating conveyor in its role as a storage and delivery device as a part of a larger system composed of a loader, the conveyor and the service channels. He develops three intuitive principles governing recirculating conveyor operation, but presents no operating characteristics other than how to calculate limiting values of speed, capacity and uniformity of loading.

Pritsker (6), 1966, using simulation methods, investigates the effect of feedback (recirculation) for systems experiencing both deterministic and Markovian arrivals, and concludes that distance between channels and feedback delay (distance) do not affect the steady-state operating characteristics of the system. A careful reading of Pritsker's investigation will reveal, however, that he (1) precludes the formation of an arrival queue at the loading point, (2) permits the possibility for recirculated items and newly arrived items to arrive at the first server at the same virtual instant in real time, and (3) assumes zero time delay for items transiting between channels in search of an open channel, while simultaneously specifying a finite time delay on the return portion of the conveyor. These are the characteristics, not of a constant-speed system, but rather of a looped system, for example, composed of two belt-type conveyors (one, the supply conveyor to the servers, running very fast -- perhaps at "infinite" speed to provide the zero time delay between servers -- and the other, the return conveyor, running at some slower, finite speed), with both belts being randomly loaded (i.e., with loads randomly spaced).

The same comments may be directed to the simulations used by Phillips (4), 1969, and Phillips and Skeith (5), 1969, who extend Pritsker's simulation to include more system operating characteristics. The essential difference between Phillip's (4) model and Pritsker's (6) is that Phillips requires a queue to build at the head of the return conveyor if the return conveyor is occupied, whereas Pritsker's model does not contain this restriction.

These prior investigations apparently fail to model the constant-speed recirculating conveyor faithfully; that is, a recirculating conveyor in the form of a continuous loop that operates at constant speed. Consequently, the effects due to finite time delays occuring in the system are not isolated and analyzed. Thus, a remedy for these deficiencies became the impetus for the development of the model described herein.

An additional objective, adopted early in the conceptual phase, was to create a utility simulation model of the constant-speed recirculating system -- a model that would be highly flexible and capable of application over a wide range of operating conditions without changing the program code. The result is presented herein.

Description of the Physical System

The simulation program presented in this paper models the physical system depicted in Figure 1. This system consists of a supply point (or "loader"), the recirculating closed-loop conveyor, and the m service facilities or channels ("servers").

Functionally, the system shown in Figure 1 can be described as follows. Semi-processed items, presumably from a prior stage in a manufacturing process, arrive at the tail of the loader at random times, according to some inter-arrival time distribution with a mean rate λ. If a queue is present in the loader, an arriving item waits; otherwise, it is loaded directly on the first empty position on the passing conveyor. Queue discipline in the loader is FIFO. Each such on-loaded arriving item, or any recirculated item, is conveyed at constant conveyor speed to the first service channel for additional processing, where the server is polled for availability. If the first server is idle, or if space in a local storage reserve is available, then the conveyor-supplied item is off-loaded and either enters the server or is added to the local storage of reserved (useables), as the case may be. If the first server or the first storage is full, the conveyor-supplied item remains on the conveyor and, a short time later, depending upon the constant conveyor speed, polls the second server for availability. If the second server is available (idle server or non-full storage reserve), the item is off-loaded; otherwise it continues to the third server, and so forth.

The polling sequence is repeated until one of two events occur: (a) entry is gained at one of the subsequent channels, or (b) entry is not gained at any channel, causing the conveyor-supplied item to recirculate at constant speed on the return portion of the conveyor, finally to reappear again at the first channel where the polling sequence begins again. With the proper choice of system parameters, in the form of arrivals, services, conveyor speed, and physical dimensions of the system, the arrivals at the loader constitute a "birth" process and the services in the servicing channels constitute a "death" process, so that the system can be operated at stochastic equilibrium.

The conceptual system shown in Figure 1 is based on the following assumptions:

1. Arrivals into the system (at the tail of the loader queue) are assumed to follow an interarrival time distribution of the investigator's choice (e.g., exponential, deterministic, etc.)

2. Local storage (reserve) may or may not be specific at any channel -- that is, the capacity of each server is restricted to the item receiving service, plus any value of local storage (including zero).

3. All service channels have the same mean service rate.

4. Outputs from the service channels are not returned to the recirculating conveyor.

5. The unloading of an item from the conveyor is determined solely on the basis of a free channel on local storage availability; that is, the unloading decision is made at the discrete point in time when the item reaches a particular channel (no "look-ahead"). Items poll the servers for availability in sequence in the direction of conveyor movement, and items failing to gain admission at the m^{th} (last) channel are recirculated.

6. The "hooks" on the conveyor are equally spaced and the conveyor moves at constant speed. Since the conveyor is a continuous loop, both service and return portions move at the same speed.

7. The physical dimensions of the system -- that is, loader to first channel distance, return conveyor length, and inter-channel spacing -- are finite ($0 < D_L, D_R, D_S < \infty$).

Model Description

Based on the foregoing assumptions and on the functioning of the physical system, a simulation model was encoded in the GPSS/360 simulation language (7).

The flow chart in Figure 2 displays the basic coding. The format of Figure 2 generally follows the physical system in Figure 1 to facilitate identification of the coded program steps with the functional portions of the system in Figure 1. Most of the GPSS statistical tabulation blocks (the system "instrumentation" by which output information is gathered) have been omitted from Figure 2 for clarity.

In describing the simulation model the following topics will be presented: composition, model input, model functioning, unique features, model output, and model execution. Each of these topics is discussed in the following paragraphs.

Model Composition

Referring to Figure 2, the model system program consists of eight functional portions: (1) data input cards (not illustrated), (2) a data initialization sequence (not illustrated), (3) a sequence of blocks representing the recirculating conveyor, (4) a sequence of blocks providing for system arrivals and representing the loader, (5) a generalized block sequence representing from 1 - 10 service channels, (6) a block sequence representing the unloading operation at the service channels, (7) a data computing sequence (not illustrated), and (8) a report generator (not illustrated).

By means of the REALLOCATE feature in the GPSS language best use is made of the necessary core, so that the program in its present version runs in 122K of a 128K partition on the IBM 360 series computers. Compile and run time together on a 360/65 varies from about 40 seconds to more than 2 minutes, depending upon the "length" of the conveyor (total number of transactions in the system) and on the number of service channels (proportional to the amount of internal computation required). The entire source deck including comment cards is about 300 cards in length.

Model Input

Two distinguishing features of a utility simulation program are flexibility of application and ease with which the model configuration can be changed. These objectives are accomplished in the model by writing the program in generalized form and INITIALizing input data. All input to the program is through ten INITIAL cards, one STORAGE card, and two FUNCTION definitions. By means of these inputs, the following system parameters are controlled:

1. Number of service channels (1 - 10),
2. Spacing between channels (in "hook" pitches),
3. Distance from loader to first channel (pitches),
4. Total length of conveyor (no. of hooks),
5. Conveyor "hook" pitch (in feet),
6. Conveyor speed (feet/min),
7. Mean interarrival time of arrivals at the loader queue tail (millimin),
8. Mean service time of a served item (millimin),
9. Standard deviation of interarrival time,
10. Standard deviation of service time,
11. Capacities of local storage ("useables") at each service channel (no. of items),
12. Statistical distribution of item interarrival time,
13. Statistical distribution of item service time.

These system parameters completely specify any constant-speed recirculating conveyor-supplied system of the form depicted in Figure 1, in which loads are spaced at random discrete intervals (by integral multiples of the "hook" pitch). By merely specifying the values of these parameters, one can use this model without change to simulate any particular system of the type mentioned which incorporates any arrival distribution and any service time distribution (given that they have finite variances), and which consists of 400 or less "hooks" and 10 or less service channels.

Model Functioning

After the necessary data are read in to describe the exact model configuration desired, the GPSS program first generates one transaction which performs some calculations on these data and stores the results in SAVEVALUEs for subsequent internal use. (These details are not reported here).

Subsequently, the next block sequence of interest is the manner in which the program "builds" the recirculating conveyor and simulates its operation (see Figure 2). The "hook" transactions representing the conveyor are generated by the "conveyor building generator":

```
GENERATE  X30,,2,X5,0,6,F
```

SAVEX 30 contains the mean creation time which the program has earlier computed from the specified "hook" pitch and the conveyor speed. No modifier is specified in Field B, so that the conveyor "hook" transactions are created at constant intervals of the mean time required by SAVEX 30. The first of the "hook" transactions is created at an offset of 2 time units (Field C), to permit input data initialization and calculation to take place via another (earlier) transaction originating in a preceding GENERATE block (not depicted in Figure 2). SAVEX 5 (Field D) contains the total number of transactions (number of "hooks") to be generated, each of which is generated with PRIORITY 0 (Field E), and also with 6 fullword parameters (Fields F and G). In this manner, the total length of the conveyor is established, as well as the discrete (and uniform) time spacing of each "hook" corresponding to the conveyor speed and pitch.

Since each "hook" transaction in the entire circuit is never destroyed and always experiences the same relative incremental delays in the circuit, each "hook" transaction retains its identity throughout the simulation and also its time-relative position to every other "hook" at all times, regardless of its physical location in the system logic. The functional effect is thus one of a specific sequential, time-ordered, finite numbered set of transactions which behave in the simulation as a conveyor chain on which load points ("hooks") are discretely spaced and which move at constant speed in a closed loop.

The six "hook" transaction parameters are utilized as follows:

P1: Contains the value 0 or 1, to indicate whether the "hook" is loaded (1) or empty (0).

P2: Contains a value (1, 2, ..., 10) to indicate the next service channel to be polled for entry. If entry is successful, this value becomes the channel identifying index.

P3: Contains a value for the variable time delay to be experienced by the "hook" transaction in "skip" transiting between a channel where a load has just been unloaded and the beginning point of the recirculation loop.

P4: Contains the number of recirculations experienced by a loaded "hook" before it is unloaded (reset to zero immediately after the "hook" is unloaded).

P5,P6: For use in future modifications of the model.

The next block sequence of interest is the one defining the operation of the "loading station" and its interaction with the conveyor "hook" transactions. The transactions representing incoming physical items to be conveyed and processed by the system are generated by the block

```
GENERATE  X7,FN1,X25,,1,2,F  .
```

SAVEX 7 contains the "item" mean interarrival time (from INITIAL data), which is modified by FUNCTION 1, the interarrival time distribution (any desired normalized statistical distribution with finite variance). The offset interval is specified in Field C, which uses the value contained in SAVEX 25 -- a value calculated internally in the data initialization sequence, sufficiently great timewise to permit the recirculating conveyor to be "built" before the first item to be processed "appears" at the loader. There is no creation limit (Field D), and each "item" transaction is created with PRIORITY 1 (Field E) and with 2 fullword parameters (Fields F and G).

The key blocks in the "loader" are the ENTER MAIN 1, GATE LS 1, and LEAVE MAIN 1 blocks. Storage "MAIN" is capacitated by a STORAGE definition card to a value of one. GATE LS 1 prevents the forward movement of an "item" transaction in "MAIN" until LOGIC Switch 1 is set ("on"). This combination of blocks forces the formation of a first-come, first-served queue ("LINE") of "items" behind the loader in the event the conveyor cannot load "items" as fast as they arrive.

Assuming now that the simulation has advanced to the point where there is an "item" transaction in "MAIN" (delayed by GATE LS 1), and where a queue ("LINE") exists waiting to enter "MAIN", the interaction of the conveyor with the loader can be described as follows. A recirculating "hook" transaction is first tested at

```
LOAD  TEST E     P1,0,BYPAS
```

to determine if the conveyor "hook" is empty (P1 = 0). If it is already loaded (P1 ≠ 0) the "hook" bypasses the loading sequence to the BYPAS ADVANCE X26 block. If the "hook" is empty (P1 = 0), it then tests for the presence of one or more "item" transactions in the loader:

```
TEST G     Q$LINE,0,BYPAS
```

If there is no item waiting (Q$LINE = 0), the "hook" bypasses to BYPAS ADVANCE X26. If there is an item waiting (Q$LINE ≠ 0), the "hook" transaction then sets LOGIC S 1 ("on") and immediately encounters the BUFFER block, with the following consequences:

(a) the current events chain scan is re-initiated,

(b) the "hook" transaction is delayed with zero relative time change (since it is Priority 0),

(c) the "item" transaction in "MAIN" is activated and moves (since it is Priority 1) as a result of Logic Switch 1 having been set ("on") by the "hook" transaction. The "item" transaction then continues to move, which resets Logic Switch 1 (resetting the "gate" so as to delay a subsequent "item" transaction), then on to TERMINATE 0 after passing through the DEPART LINE, TABULATE, an LEAVE MAIN 1 blocks.

This sequence destroys one "item" transaction, which has now served its purpose -- representing a load or "item" for the "hook" waiting in BUFFER. After the "item" transaction is TERMINATEd, the waiting "hook" transaction moves immediately (since it was delayed with zero relative time), and its Parameter 1 is changed from 0 to 1 (signifying now a loaded "hook") by the ASSIGN 1, K1 block. The "loaded hook" transaction then proceeds to the BYPAS ADVANCE X26 block, where it is delayed for the transit time (X26) analogous to the conveyor distance from the loader to the first service channel. (The time delay value in SAVEX 26 is internally calculated once from the initialized input data by an earlier one-time transaction, and all "hooks", loaded or unloaded, are delayed by the same increment of relative time).

The next block sequences of interest are the "channel test sequence" and the "channel indexing sequence" (Figure 2). These are nearly self-explanatory, except to point out that the first test (CNT2 TEST E P1,1,ADV31) is to determine if the "hook" transaction is loaded (P1 = 1). If not, the hook bypasses the channel polling sequence to the ADVANCE block (ADV31 ADVANCE X31) which provides the time delay necessary (stored in SAVEX 31) to transit the distance between the first and last service channels. Otherwise, if the "hook" is loaded (P1 = 1), the ASSIGN 2,K1 block assigns the value 1 to Parameter 2, which indicates that the polling sequence is to begin with Channel 1.

Channel 1 is tested for entry in the block

GATE1 GATE SNF P2,NEXT1

(when the value of the channel index, P2 = 1), which allows the "loaded hook" transaction to enter the "unloading sequence" at Channel 1 if that channel (or its local storage) is not full; otherwise the transaction moves to the block

NEXT1 TEST L P2,X1,OVER

which determines if the last channel has been tested. If not, the transaction moves through the indexing sequence in which the channel identifying number (P2) is incremented by one, a delay is encountered corresponding to the distance between two adjacent channels, and the next sequential channel is polled for entry. Loaded, unserviced "hook" transactions, after polling each channel and failing to unload, then begin the recirculation journey via the ADVANCE X28 block after passing through the statistical TABULATE RECIR block, which tabulates the recirculation frequency of loaded "hooks".

If a "loaded hook" finds an open service channel or available local storage in the polling sequence, it then enters the "unloading sequence" blocks (Figure 2). Here, the entering "hook" transaction is first increased in priority (PRIORITY 2 block), so that when a duplicate or "copy" transaction (representing an unloaded "item") is created in the SPLIT 1, CHAN block, the duplicate will have a higher priority in the current events chain. Immediately after the SPLIT block, the "hook" transaction encounters the PRIORITY 0, BUFFER block, which develops the following consequences:

(a) the priority of the parent "hook" transaction is reduced (PR = 0), while the priority of the duplicate transaction (representing the unloaded "item") remains higher (PR = 2);

(b) the current events chain scan is reactivated by the BUFFER command;

(c) the "copy" transaction moves first (due to PR = 2), ENTERing the storage whose identifying channel number is contained in Parameter 2 of the "copy" transaction, joining the appropriate QUEUE*2, and so forth;

(d) when the "copy" transaction ceases to move, due to QUEUE*2 or ADVANCE X8,FN2, then the parent "hook" transaction (now with PR = 0) moves again through the remainder of the unloading sequence blocks, which perform the housekeeping functions noted in Figure 2. It is noteworthy to add that the ASSIGN 1, KO block changes the value of Parameter 1 in the "hook" transaction from 1 to 0, signifying that the "hook" has now been unloaded; thus, after the recirculation trip it is again available for loading again at the loader.

The final group of blocks of particular interest in Figure 2 is the "generalized service channel sequence", which models from one to ten service channels and their associated local storages. These blocks are fairly self-explanatory, although it should be noted that one of the outstanding utility features of the entire model is embodied in this sequence. To write the storage, queue and service facilities in general form, we use to advantage the fact that the channel, storage and queue identifying number is contained in Parameter 2 of the "copy" transaction created in the SPLIT block. Thus, one can use this identifying number in the form *2 to identify which ENTER, QUEUE, SEIZE, DEPART, RELEASE, and LEAVE blocks and statistics are to be affected by a particular "item" unloaded from the "hook". It should also be noted briefly that we use Parameters 3 and 4 of the "copy" transactions for the accumulation of statistical data concerning waiting and service times of an "item" being serviced, rather than for the purposes mentioned earlier for the parent "hook" transaction.

Unique Features

Some of the unique features of the model, described in detail above, can be summarized as follows:

(1) the GENERATion of a finite number of sequenced, undestroyed transactions, equally spaced in relative time, to represent a closed-loop conveyor with equally spaced load points ("hooks") travelling at constant speed throughout the simulation;

(2) the interfacing of random "item" arrivals with the continuously moving conveyor at the "loader", which realistically models the conditions (a) that an "item" is "loaded" only when an empty conveyor "hook" is available, and (b) that arriving "items" build a first-come-first-served queue if they arrive momentarily faster than the conveyor can "load" them;

(3) as a consequence of (2), "items" arrive at the first service channel at random integral multiples of the time required for the conveyor to advance one pitch length -- this is different from the completely random interarrival times assumed by prior investigators for belt-type conveyors;

(4) the "generalized service channel sequence" that permits representation of up to 10 service channels by one set of blocks;

(5) all input to the program, completely specifying the model configuration and operation, is through the use of the INITIAL, STORAGE and FUNCTION cards, which makes major model changes necessary in order to simulate different systems.

Model Output

The output from the simulation occurs via the REPORT editor, and consists of the following:

1. Report of all data inputs.
2. Report of internally calculated data:
   - (a) Conveyor length (feet)
   - (b) Distance between channels (feet)
   - (c) Distance, loader to first channel (feet)
   - (d) Distance, first to last channel (feet)
   - (e) Distance of return loop (feet)
   - (f) Conveyor circuit transit time (min)
   - (g) Transit time, loader to first channel (min)
   - (h) Transit time, first to last channel (min)
   - (i) Transit time, return loop (min)
   - (j) Transit time, between channels (min)
   - (k) Transit time, one pitch length (min)
   - (l) Conveyor service rate (hooks/min)
   - (m) Nominal input utilization (new

arrivals) on supply portion of conveyor (decimal)

(n) Mean input utilization (new arrivals) on supply portion of conveyor (decimal)

(o) Mean utilization (recirculated items) on return loop of conveyor (decimal)

(p) Mean total utilization (new arrivals plus recirculated items) on supply portion of conveyor (decimal)

(q) Mean frequency of recirculation, given that an item recirculates on return loop (decimal)

(r) Probability of recirculation (= relative frequency with which all items recirculate, including those that do not)

(s) Nominal overall system utilization (overall "black-box" utilization), %.

3. Utilization of all service channel facilities.
4. Utilization and other storage statistics for all storages, including local storages at each channel.
5. Queue statistics for each queue in the model.
6. The statistical distributions of:

(a) Recirculation frequency of items that recirculate once or more;

(b) Waiting times in each service channel queue;

(c) Service times for each service channel;

(d) Departure rate from each service channel;

(e) Waiting time in the "loader" queue;

(f) Loading rate (pick-up rate by the conveyor) at the "loader";

(g) Loaded hook recirculation rate on the return loop;

(h) Loaded hook arrival rate at the first service channel; and

(i) Inter-item times corresponding to the rate distributions mentioned in (f), (g) and (h).

The richness of these selected performance statistics permits almost any post-simulation operational analysis to be made, as will be seen in the sample simulation results reported in subsequent paragraphs.

Model Execution

The program has been used so far to simulate about 80 different conveyor system configurations, ranging from 1 to 4 channels and with various loop-lengths, various distances between loader and first channel, various channel spacings, various speeds, "hook" pitches and other operating parameters. For a typical simulation, the following simulation control cards would be used:

```
*
*
 START        200,NP
 RESET
 START        1000,NP
 GENERATE     ,,,1,,0
 SAVEVALUE    9,V9
 SAVEVALUE    12,V12
      .
      .
      .
 SAVEVALUE    17,V17
 TERMINATE    1
 START        1
 REPORT
*
*
```

The START 200,NP command initially "loads" an empty and idle system which, in response to the RESET card, permits the actual simulation to commence with the START 1000,NP command. This tends to remove the transient start-up effect from the output statistics, which is desirable for this model. Realistically, a recirculating conveyor in a manufacturing plant, for example, would be used continuously or if it were stopped intermittently (say, overnight) then it would be restarted in the same loaded condition as when it was stopped. Thus, the steady-state operating condition is of primary interest, not the transient phase.

Following the START 1000,NP card which initiates the actual simulation, a GENERATE block is used to generate one final transaction whose function it is to calculate some statistical information from data developed in the simulation. The statistical information is stored in SAVEVALUES, via numerical calculations performed by VARIABLE statements analogous to the arguments of the SAVEVALUE blocks. Once this duty is performed the final transaction is destroyed in the TERMINATE 1 block, which ends the simulation. The REPORT card initiates the GPSS report editor, and output ensues.

A comment should be made about the transaction sample sizes (200 and 1000) used to "load" the system and perform the simulation. These are merely representative. In an actual simulation, one would need to replicate the experiment with a particular configuration several times with different random number seeds (using the RMULT feature), so as to provide an independent measure of the random error in the experiment. Then statistical tests for significance could be applied to the experimental results, and the experimenter could decide whether or not his simulation sample were large enough for steady-state to have been reached.

Typical Simulation Results

As a result of some prior work with this model, several interesting and unexpected phenomena have been observed in studying constant speed recirculating conveyors that are random discretely loaded and serviced.

For example, consider the following:

1. Based on an argument that finite delay times in the system give rise to opportunities

for system regeneration points to occur (which would not be possible if delay times were zero), it was hypothesized that system performance parameters in the finite delay system would be different from those given by earlier investigators who assumed zero time delays in the system.

The authors' earlier research substantiated this hypothesis (1). Two system performance parameters, the probability of recirculation, P_r, and the return conveyor utilization, ρ_R, were shown to be functions not only of the system loading, ρ_s, but also the input loading, ρ_i, the latter being related inversely to conveyor speed (see Figure 3). If these performance parameters were independent of finite delay times, then they would be invariate with changes in conveyor speed; however, such was not the case. Both of these performance parameters were shown to be non-simple functions of finite delay times in the system (Figure 3). This confirmed the hypothesis and also tended to indicate that an assumption of zero delay times within or between points of the system is not a realistic one for constant-speed finite-delay recirculating systems.

2. Another result is that constant-speed (finite delay) recirculating conveyor-supplied systems can display "blocking" on the service (supply) portion of the conveyor (see also Figure 3), a phenomenon not possible under prior research in which zero time delays were assumed. A constraint of all such finite delay recirculating systems is that they be operated so that the utilization of the service conveyor, ρ_c^*, never exceeds unity (see also Muth (3) for a different approach with the same conclusion for cyclically-loaded continuous conveyors). This constraint apparently does not exist and has not been reported for previously investigated recirculating conveyors.

3. It can also be inferred that the probability of recirculating, P_r, collapses into two theoretical conditions as "end points", when conveyor speed (or input loading, ρ_i) reaches limiting values (see Figure 4). For the case in which conveyor speed becomes "infinitely" large and the number of hooks great compared to a possible queue length, the system is analogous to a zero delay search condition, and P_r apparently collapses into theoretical values for P(m), the probability of all channels simultaneously busy, for a theoretical $M/M/m:\infty$ queueing system. For the single server system case when the conveyor speed is slowed so that blocking occurs on the serving conveyor ($\rho^*_c = 1.0$), arrivals at the service channel become exactly deterministic, and P_r apparently collapses into theoretical values of P(1) for the theoretical D/M/1:0 queueing system. In between these limiting cases the probability of recirculation (analogous to probability of "overflow" for a non-feedback system) takes on intermediate values indicative of a complex function of the finite delay experienced between channels, the form of which is not

presently understood.

4. A final result is that constant-speed recirculating conveyors not only can be used as storage devices, but that maximum utilization of the return leg effectively occurs in the approximate range $0.5 \leq \rho_i \leq 0.7$ for all values of m (number of servers) and ρ_s (system loading) -- see Figure 3. In this range, the expected loader queue length is small (virtually all arrivals are "stored" on the conveyor). Thus, since the input loading, ρ_i, is under control of the designer, an opportunity exists for an economic trade-off of queue storage space at the loader for increased conveyor speed or decreased hook spacing, the principal controllable determinants of ρ_i.

Conclusions

The GPSS/360 simulation program described herein effectively models the operation of a constant speed recirculating conveyor-supplied multiserver system, in which random arrivals are accommodated and served, and in which the conveyor is loaded in a random discrete manner with load-points ("hooks") equally spaced on the conveyor loop.

The program is effective as a utility program for the investigation of different system configurations without the necessity of changing the model. To date, over 80 different configurations have been investigated, with the sole changes in the program being in the 10 INITIAL cards, 1 STORAGE card, and 2 FUNCTION definitions.

Compilation and execution time for a simulation is quite short considering the logic complexity of the system -- on the order of 40 seconds to 2.5 minutes on an IBM 360/65 computer.

The value of this utility program is demonstrated by the fact that preliminary usage has already provided some interesting and unexpected operating phenomena concerning discretely loaded recirculating constant-speed conveyors, not heretofore observed quantitatively; namely, the "blocking" phenomenon, the dependence of certain system performance parameters on loadings and conveyor speed, and the apparent collapse of the probability (relative frequencey) of recirculation into theoretically determined values under limiting conditions of system operation.

References

1. Bussey, Lynn E., and M. P. Terrell,"Characteristics of Constant-speed Recirculating Conveyors with Markovian Loadings and Services", presented at 39th National Meeting, ORSA, Dallas, May 3-9, 1971.
2. Kwo, T. T., "A Theory of Conveyors", Management Science, Volume 6, Number 1, Pages 51-71, 1958.
3. Muth, Eginhard, "System Engineering Applied to the Analysis of Continuous Flow Conveyors", Technical Report No. 64, Department of Industrial and Systems Engineering, University of Florida, Gainesville (1972).
4. Phillips, Donald T., "A Markovian Analysis of the Conveyor-Serviced Ordered Entry Queueing System with Multiple Servers and

Multiple Queues", (Unpublished Ph.D. dissertation, the University of Arkansas, Fayetteville, 1969).

5. Phillips, Donald T. and Ronald W. Skeith, "Ordered Entry Queueing Networks with Multiple Servers and Multiple Queues", AIIE Transactions, Volume 1, Number 4, Pages 333-342, December, 1969.

6. Pritsker, A. Alan B., "Application of Multichannel Queueing Results to the Analysis of Conveyor Systems", Journal of Industrial Engineering, Volume 17, Number 1, Pages 14-21, January 1966.

7. "General Purpose Simulation System / 360 -- User's Manual", Publication No. H20-0326-2, (International Business Machines Corp., White Plains, New York, Technical Publications Department, 1968).

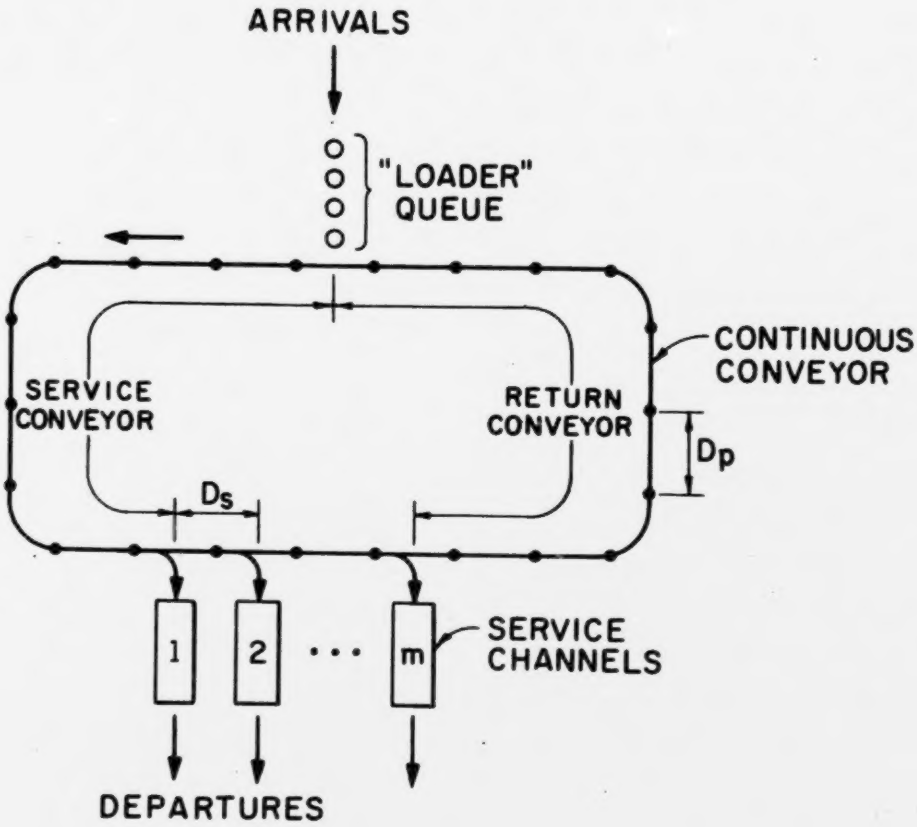

ARRIVALS
"LOADER" QUEUE
CONTINUOUS CONVEYOR
SERVICE CONVEYOR
RETURN CONVEYOR
Dp
Ds
1
2
m
SERVICE CHANNELS
DEPARTURES

Figure 1 - Conveyor-Supplied Multiserver System.

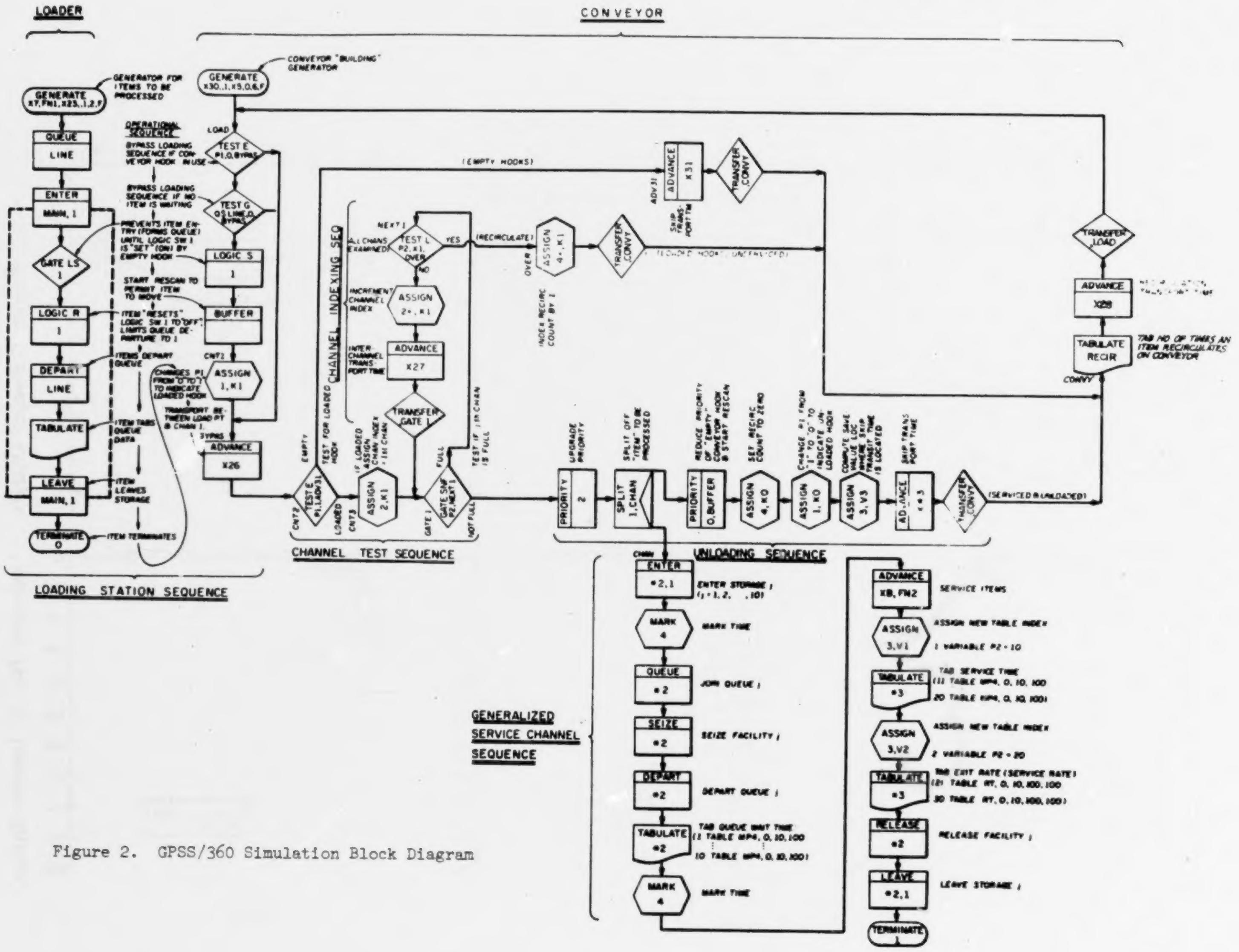

LOADER
CONVEYOR
GENERATE X7,FN1,X23,,1,2,F
GENERATOR FOR ITEMS TO BE PROCESSED
QUEUE LINE
ENTER MAIN,1
GATE LS 1
LOGIC R 1
DEPART LINE
TABULATE
LEAVE MAIN,1
TERMINATE 0
OPERATIONAL SEQUENCE
BYPASS LOADING SEQUENCE IF CONVEYOR HOOK IN USE
BYPASS LOADING SEQUENCE IF NO ITEM IS WAITING
PREVENTS ITEM ENTRY (FORMS QUEUE) UNTIL LOGIC SW 1 IS "SET" (ON) BY EMPTY HOOK
START RESCAN TO PERMIT ITEM TO MOVE
ITEM "RESETS" LOGIC SW 1 TO "OFF", LIMITS QUEUE DEPARTURE TO 1
ITEMS DEPART QUEUE
ITEM TABS QUEUE DATA
ITEM LEAVES STORAGE
ITEM TERMINATES
CHANGES P1 FROM "0" TO "1" TO INDICATE LOADED HOOK
TRANSPORT BETWEEN LOAD PT & CHAN 1.
GENERATE X30,,1,X5,0,6,F
CONVEYOR "BUILDING" GENERATOR
LOAD
TEST E P1,0,BYPAS
TEST G Q$LINE,0,BYPAS
LOGIC S 1
BUFFER
CNT1
ASSIGN 1,K1
BYPAS
ADVANCE X26
LOADING STATION SEQUENCE
(EMPTY HOOKS)
CHANNEL INDEXING SEQ
NEXT1
ALL CHANS EXAMINED
TEST L P2,X1,OVER
YES
NO
INCREMENT CHANNEL INDEX
ASSIGN 2+,K1
INTER-CHANNEL TRANSPORT TIME
ADVANCE X27
TRANSFER GATE1
CNT2
EMPTY
TEST E P1,1,ADV31
TEST FOR LOADED HOOK
LOADED
CNT3
IF LOADED ASSIGN CHAN INDEX = 1st CHAN
ASSIGN 2,K1
GATE1
FULL
GATE SNF P2,NEXT1
TEST IF 1st CHAN IS FULL
NOT FULL
CHANNEL TEST SEQUENCE
(RECIRCULATE)
OVER
ASSIGN 4+,K1
INDEX RECIRC COUNT BY 1
TRANSFER ,CONVY
(LOADED HOOKS UNSERVICED)
ADV31
ADVANCE X31
SKIP TRANSPORT TIM
TRANSFER ,CONVY
TRANSFER ,LOAD
ADVANCE X28
TABULATE RECIR
TAB NO OF TIMES AN ITEM RECIRCULATES ON CONVEYOR
CONVY
UPGRADE PRIORITY
PRIORITY 2
SPLIT OFF "ITEM" TO BE PROCESSED
SPLIT 1,CHAN
REDUCE PRIORITY OF "EMPTY" CONVEYOR HOOK & START RESCAN
PRIORITY 0,BUFFER
SET RECIRC COUNT TO ZERO
ASSIGN 4,K0
CHANGE P1 FROM "1" TO "0" TO INDICATE UNLOADED HOOK
ASSIGN 1,K0
COMPUTE SAVE VALUE LOC WHERE SKIP TRANSIT TIME IS LOCATED
ASSIGN 3,V3
SKIP TRANSPORT TIME
ADVANCE *3
TRANSFER ,CONVY
(SERVICED & UNLOADED)
UNLOADING SEQUENCE
CHAN
ENTER *2,1
ENTER STORAGE i (i = 1, 2, ... ,10)
MARK 4
MARK TIME
QUEUE *2
JOIN QUEUE i
SEIZE *2
SEIZE FACILITY i
DEPART *2
DEPART QUEUE i
TABULATE *2
TAB QUEUE WAIT TIME (1 TABLE MP4, 0, 10, 100 ⋮ 10 TABLE MP4, 0, 10, 100)
MARK 4
MARK TIME
ADVANCE X8,FN2
SERVICE ITEMS
ASSIGN 3,V1
ASSIGN NEW TABLE INDEX
1 VARIABLE P2+10
TABULATE *3
TAB SERVICE TIME (11 TABLE MP4, 0, 10, 100 ⋮ 20 TABLE MP4, 0, 10, 100)
ASSIGN 3,V2
ASSIGN NEW TABLE INDEX
2 VARIABLE P2+20
TABULATE *3
TAB EXIT RATE (SERVICE RATE) (21 TABLE RT, 0, 10, 100, 100 ⋮ 30 TABLE RT, 0, 10, 100, 100)
RELEASE *2
RELEASE FACILITY i
LEAVE *2,1
LEAVE STORAGE i
TERMINATE 1
GENERALIZED SERVICE CHANNEL SEQUENCE

Figure 2. GPSS/360 Simulation Block Diagram

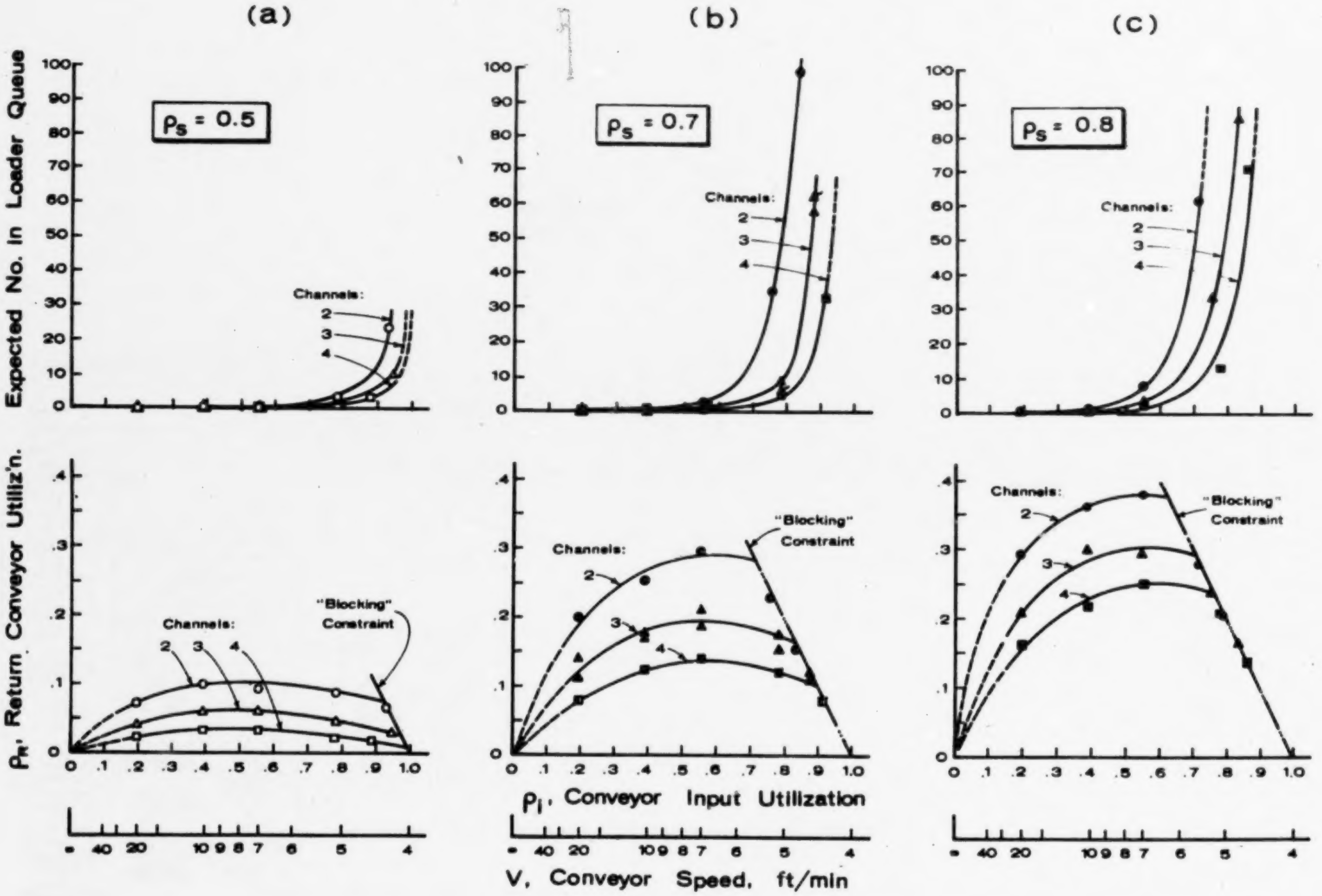


Figure 3. Effects of Conveyor Input Utilization, Conveyor Speed, and System Traffic Intensities on Basic Operational Characteristics.

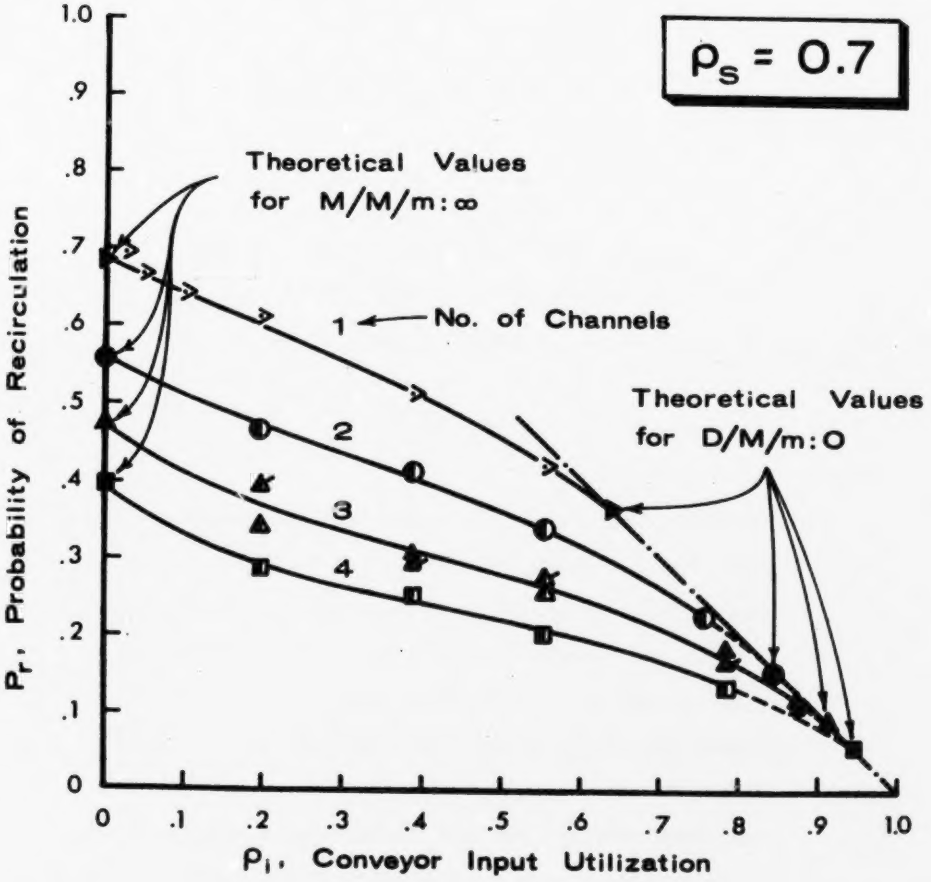


Figure 4. Effects of Conveyor Input Utilization and Number of Channels on the Probability of Recirculation.

A CASED GOODS CONVEYOR SIMULATOR

Donald A. Heimburger
Math & Computing Section
THE PROCTER & GAMBLE COMPANY
Cincinnati, Ohio

Abstract

The SIMSCRIPT Conveyor Program (SIMCON) models the flow of cased goods through an accumulating powered belt conveyor system. The program is almost totally general with all configuration descriptions and parameters being read into the model as input data. The model has been used to test proposed changes to existing systems, to design totally new systems, to study the effects of several variables, and to model new concepts in cased goods merging. The quantitative results have been used in the decision process for many multi-million dollar conveyor networks.

SUMMARY

At the Procter & Gamble Company, powered accumulating belt conveyors are used in manufacturing plants to transport high-volume, low-cost consumer cased goods. The SIMSCRIPT Conveyor Simulator (SIMCON) is an operational tool used to provide quantitative assistance in the decision making process concerning the design of these conveyors. Although an earlier conveyor simulator written in GPSS was available, SIMCON was developed because of the cost, accuracy, and model concept advantages made possible by using SIMSCRIPT. The model has been highly successful and has given quantitative direction to decisions previously "engineered" by committee. In the first three months of use SIMCON minimized the possibility of design error that could have cost upward of $500,000 (roughly 100 times its development cost) either in over-design or post startup corrections. Since that time, the program was used to justify the use of an existing system where an addition would have cost roughly $1.5 million. SIMCON can be run in either the batch mode or via conversational terminals with prompting. It is also operational on TSO without prompting.

INTRODUCTION

Multiple packing lines for cased goods are often located in a single room. This room is typically a long distance (200' - 2000') from the cased goods warehouse. Rather than a separate conveyor from each packing line to the warehouse, the cases are usually merged onto a single conveyor that carries all the cases to the warehouse. Experience has shown that even the best thought-out plans for merging cases together have occasionally resulted in massive jams and costly re-design. A method was needed to predict, before the fact, what size build-ups could be expected for particular configurations running under many sets of packing line feed rates and conveyor belt speeds.

OBJECTIVE

The objective of the study was to devise a means to provide information on queue lengths and gate utilization for cased goods conveyor networks. The system had to be easy to access and use with little or no modeling or computing knowledge. The tool was planned to be used by both designers and engineers for new systems and evaluating proposed changes to existing networks.

Because of the diversity of all the users, the program was supposed to be available via conversational terminals as well as batch job submission methods. The conversational version was to be extremely user-oriented and self-explanatory. Because the program was intended to be run frequently, cost was also a major consideration.

PHYSICAL SYSTEM DESCRIPTION

An accumulating powered belt conveyor system for cased goods consists of two basic elements. The first element is the packing line, and the second is the merging unit. When two packing lines feed a single merging unit, the result is a single stream of cases. Several additional lines may feed in via more merging units to form a large network.

The Packing Line

Each packing line ejects cases of a known length. The length can vary from line to line, so queues are measured in feet rather than cases. The line averages a given number of cases per minute over the course of a day, and the ejection pattern can be measured.

The Merging Unit

The merging unit is usually called a gate or a traffic cop. The name is adopted because the unit regulates case flow as a traffic cop would regulate automobile flow through an intersection. The operation of the gate can best be described by the series of Figures 1 - 4.

In Figure 1 the case A1 has just arrived on the A side of the traffic cop. The spring-loaded arm swings open as the case passes and mechanically locks the other arm from opening. In Figure 2 a case has arrived at the arm on the B side, but cannot pass until the A arm shuts. Since no more cases arrive on the A side before the spring-action return is completed, the arm swings shut and locks (Figure 3). In the meantime, another case B2 arrived at the queue on the B side. As soon as the A arm locks the B arm is free to open, and the cases B1 and B2 pass through (Figure 4). The cycle continues, each side clearing its entire queue and then releasing the other side.

Two types of major problems can occur at the gate. The first happens when one side holds an arm open indefinitely. An extremely large queue results on the opposite side. The other problem occurs when the belt speeds are too slow and the queues continue to grow indefinitely.

The Total System

The total system is made up of groups of lines and gates. Since each gate merges two lines into a single stream, there is always one fewer gate than lines. A fairly typical cased goods conveying system is shown in Figure 5. This network consists of 12 packing lines and 11 merging gates.

Every configuration consists of the same basic units (i.e., gates and lines). The only distinguishing features from one system to the next are (1) the feed rates, (2) the case lengths, (3) the arm closing times, (4) the belt speeds, and (5) the sources and exits of the gates.

When designing a new accumulating powered belt conveyor network, the first three (3) items mentioned above (feed rated, case lengths, and arm closing times) are usually known. Item 5, the actual configuration of gates and lines, is frequently limited to no more than four choices. Figures 6 and 7 show two ways of merging six lines into a single conveyor stream.

Once the alternate arrangements of gates and lines are determined, several conveyor belt speeds can be examined to see how fast the belts must travel to avoid frequent and long jams.

METHOD OF SOLUTION

Thought was given to several different approaches to the problem. One method of analyzing a particular configuration would be to build a pilot facility. However, this method is hardly practical for large networks, very costly, and does not meet all of the desired objectives.

Another approach to the problem is the use of analytical techniques. Knowing the various belt speeds and gate operations, the length of the queue at the gate could be calculated. This method was in fact useful in helping to understand the workings of a gate, but is completely deterministic ignoring the consideration of randomness of case arrival and the case-length mix downstream.

The method of analysis that overcame the objectives to the pilot facility and the analytical approach but still gave the desired results was computer simulation. Once the physical description was used to create a model, ideally any number of "what if" questions could be asked to see the effects of each of the operating parameters. Alternate conveyor configurations with the same case-merging requirement (Figures 6 and 7) could also be tested. Simulation was selected as the means to provide the needed answer in a short time at a low cost.

SIMULATION LANGUAGE

The conveyor model was originally written in GPSS, since it was the only simulation language available in-house at Procter & Gamble.[2] However, several distinct drawbacks soon became apparent.

The biggest problem is that GPSS is an interpretive language, and as such, cannot be compiled. Every model was a different program, even though series of MACRO's were used to describe lines and gates. This became a cost problem and also made interface with conversational facilities difficult.

The conversational program, written in RUSH*, which is also interpretive, would (1) create a file consisting of GPSS statements, (2) schedule a GPSS job using the file as input and another file as output, and (3) selectively read results from the output file. Because different size configurations would result in a different

*RUSH - a Remote Use of Shared Hardware is a trademark of Allen-Babcock Computing, Inc.

number of MARCO's, the output could be fairly small or very large.

Explaining the GPSS model to the user was difficult because the source code was very hard to follow for the laymen. Good reports had to be generated using the RUSH program because GPSS did not provide flexible report writing facilities. To be able to provide the level of accuracy necessary, the execution time became quite long and costs soared to upwards of $60-$100/simulation.

Shortly after the GPSS model was completed, SIMSCRIPT II.5Ⓣ became available in-house. Because of the problems with the GPSS model, a new model in SIMSCRIPT seemed attractive.

Some of the reasons for changing to SIMSCRIPT were as follows:

1. The physical system was easily modeled using the concept of entities, attributes, sets and events.
2. The SIMSCRIPT model, once compiled, could be used indefinitely directly from the stored machine code, hence saving from 40-80% of the cost of running each time. Only the data was used to build each separate model.
3. The core requirements were much less than with GPSS, and the total cost was reduced by 80-90%.
4. The user could easily understand the source code. Figure 8 is an example of one of the seven events used in the model.
5. The conversational interface was easy, since the data was just entered directly into a file, and the report that was desired was printed into an output file.
6. SIMSCRIPT made it possible to build into the model several tracing features that are only used when problems occur, or to trace out the step-by-step simulation to verify correct operation. To the user the trace is transparent, and no suppression cards are necessary.

THE SIMULATION MODEL

SIMSCRIPT views the world in terms of entities ("objects" in the system), attributes (describers of the "objects"), sets (groups the "objects" may be part of) and events (points in time that mark the starting and stopping of activities or changes in status).[1] The SIMSCRIPT conveyor model (SIMCON) was easy to conceptualize in these terms.

Ⓣ trademark and service mark of Consolidated Analysis Centers Inc.

Permanent Entities

Permanent entities are objects in the model that usually exist throughout the entire simulation, such as machines or loading docks. In this model there are two types of permanent entities. The quantity of each type is specified when the model is run.

The first permanent entity is the LINE. It has attributes of feed rates, case lengths, and a set (queue) in which to place the cases. The second permanent entity is the GATE. Since the GATE receives cases from two sides, the designation of A and B is used. Each GATE has attributes of a source of cases, a conveyor speed and an arm closing for each side. It also has a designation of the queue to send the case on to.

These two types of permanent entities specify and create each specific SIMCON model, linking the proper lines and gates together in the correct sequence.

Temporary Entities

Temporary entities are objects in the model that may be created or discarded at will. In SIMCON, the CASES are the only temporary entities and their attributes are their length and a serial number for tracing.

Sets

In order to accumulate statistics, the cases are entered and removed from various sets. Each GATE has three sets associated with it. These are an A.QUEUE, a B.QUEUE, and a JUNCTION. Depending on which side the CASE approaches the GATE, it is placed in the A.QUEUE or the B.QUEUE. When a CASE is removed from a queue and is passing by the arm pivot, it is placed in the JUNCTION. The CASE is then sent on to the correct queue for the next GATE, or in the case of the last GATE, it is destroyed.

Events

There are only seven events in the model. The relationship of the events is shown in Figure 9. Each event has a single argument specifying either the LINE number (in the EJECT.CASE) or the GATE number (all the other events).

The EJECT.CASE event creates a temporary entity, assigns it the correct length and serial number, and places it in the queue of the specified GATE. It then schedules an OPEN.GATE for that GATE on either the A or B side, depending on which side the CASE is placed.

The OPEN.GATE(A or B) checks for three conditions. If any condition fails, nothing transpires. First the GATE must not be busy (no CASE in the JUNCTION); second the opposite arm must not be open (or this side would still be locked); third, the queue must have at least one CASE in it to open the GATE. If all conditions are met, the CASE is removed from the queue and placed in the JUNCTION. A PASS.GATE (A or B) is then scheduled in the time it takes the CASE to move its trailing edge to the arm pivot point.

The PASS.GATE takes the CASE from the JUNCTION and sends it on to the next queue scheduling an OPEN.GATE there. If there are no more GATES, it simply destroys the CASE. If there are more CASES in this GATE'S A.QUEUE (if the last CASE came from A.QUEUE), another OPEN.GATE is scheduled immediately, and the cycle continues until the queue is empty. When the queue becomes empty, the SHUT.ARM (A or B) is scheduled in the number of seconds specified by the input data. Historically, this runs from 1-2 seconds.

The SHUT.ARM checks only to see if another CASE arrived during the swing time. If a CASE has arrived, nothing is changed, since the next CASE will also schedule a SHUT.ARM. If no CASE arrived, the OPEN.GATE is scheduled for the opposite side.

The model is allowed to run for a time specified by the user. It then generates a report, part of which is shown in Figures 10 and 11.

Assumptions

Two basic assumptions are made for the model. The first is that case acceleration is almost instantaneous. Since most cases pass back-to-back in a continuous slug, acceleration effects are almost negligible. The second assumption is that cases turning corners pass through at the same rate as their driving belt. Actual timed studies have shown that single cases turn the corners somewhat slower than if they pass straight through, but the difference varies from gate to gate. Again, if the cases are part of a slug, this is not very important.

VALIDATION

In order to test the analytic techniques employed by the model, a scaled-down physical system was constructed out of paper. Cases were moved through the system by half second intervals. The results were compared to the printout produced by the computer simulation (using the built-in trace routines). One minor change pertaining to conversion of time units had to be incorporated to produce identical results.

SIMCON was then validated on an existing network, and predicted maximum queue build-ups varied from actual build-ups by an average of less then one foot. Over 100 simulations have been made and several systems physically constructed as a result. None of these systems have produced significant deviation from the model's predictions.

It was found that when starting the simulation, downstream gates had lower utilization until the cases were flowing throughout the entire system. For this reason the model allows cases to flow for a user-specified warm-up period, clears all statistics, and continues from that point.

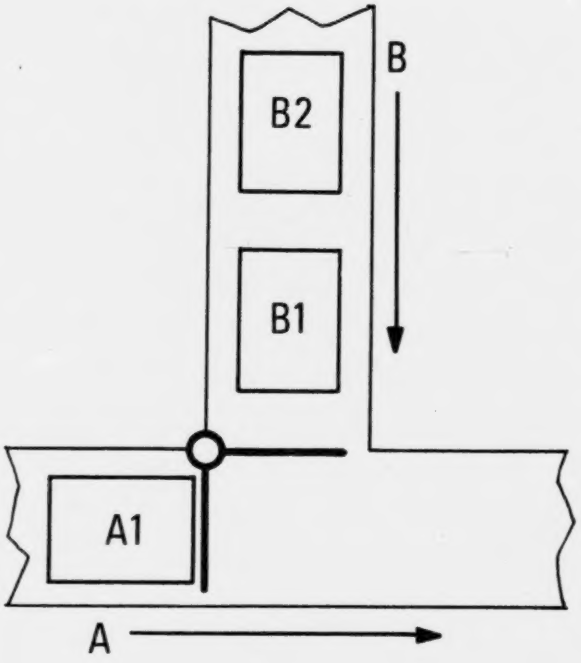
DISCUSSION

Upper management felt from the very beginning of this project that close coordination and communication were very important between the analytical team constructing the model and the intended users (Finished Products Handling Group). Both written and verbal contact were maintained through every step of development, testing and implementation; no unilateral decisions were made without consultation between both groups. As a result, all persons concerned with the project felt joint responsibility and accomplishment for the planned goals and solution. Acceptance has been 100%, and SIMCON is used as standard practice on every significant conveyor project.

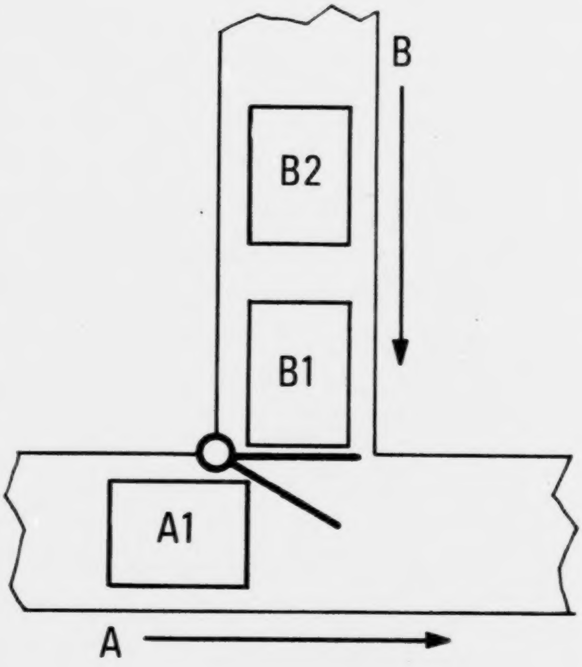
The design and implementation of the conveyor simulator was forecast to require 1.5 man-months of analytical effort and roughly $1500 for computer charges. The total development cost was within that estimate. As previously stated, within three months the program minimized the possibility of design error that could have cost upwards of $500,000 in either over-design or post start-up corrections.

SIMCON has been in use for less than one year, but already has been the primary quantitative tool for making decisions on several powered conveyor configurations costing upwards of $500,000 with the largest being roughly $3,500,000. The program is often run in the batch mode to evaluate many alternatives in a short time. It is also run from many different points in the country via conversational terminals to test effects of changes to existing networks.

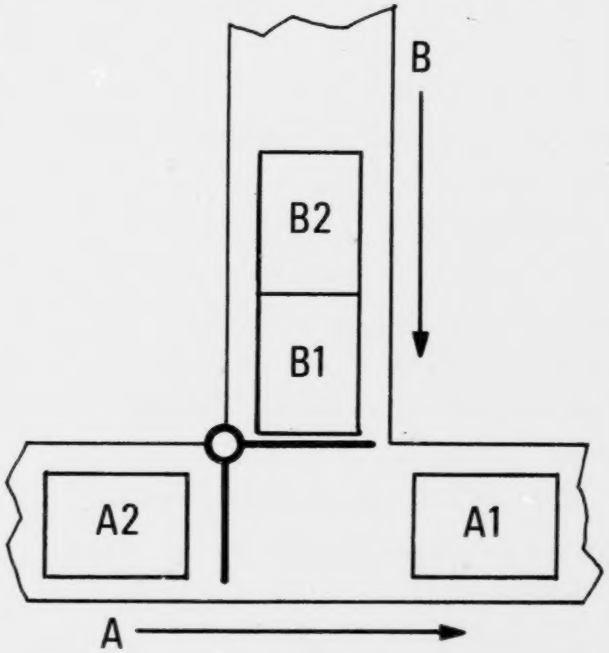
Recompilation of the program has not been necessary since it was installed and released for use.



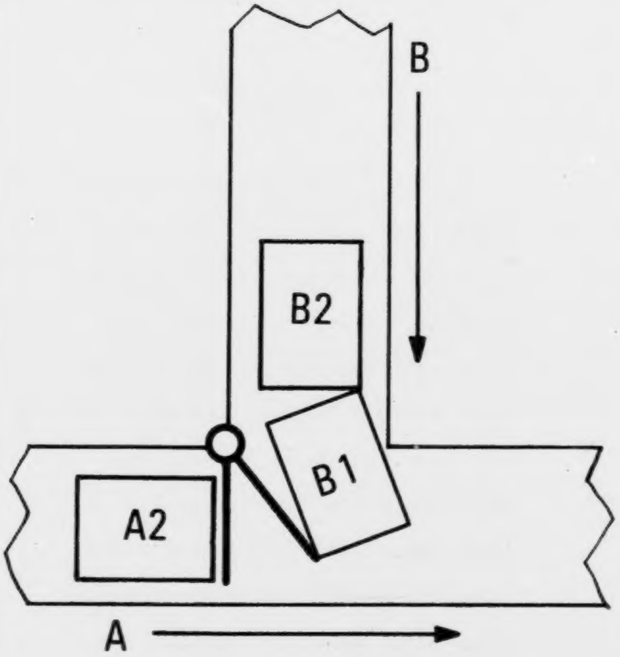
CASE A1 ARRIVES, A.ARM OPENS
Fig. 1



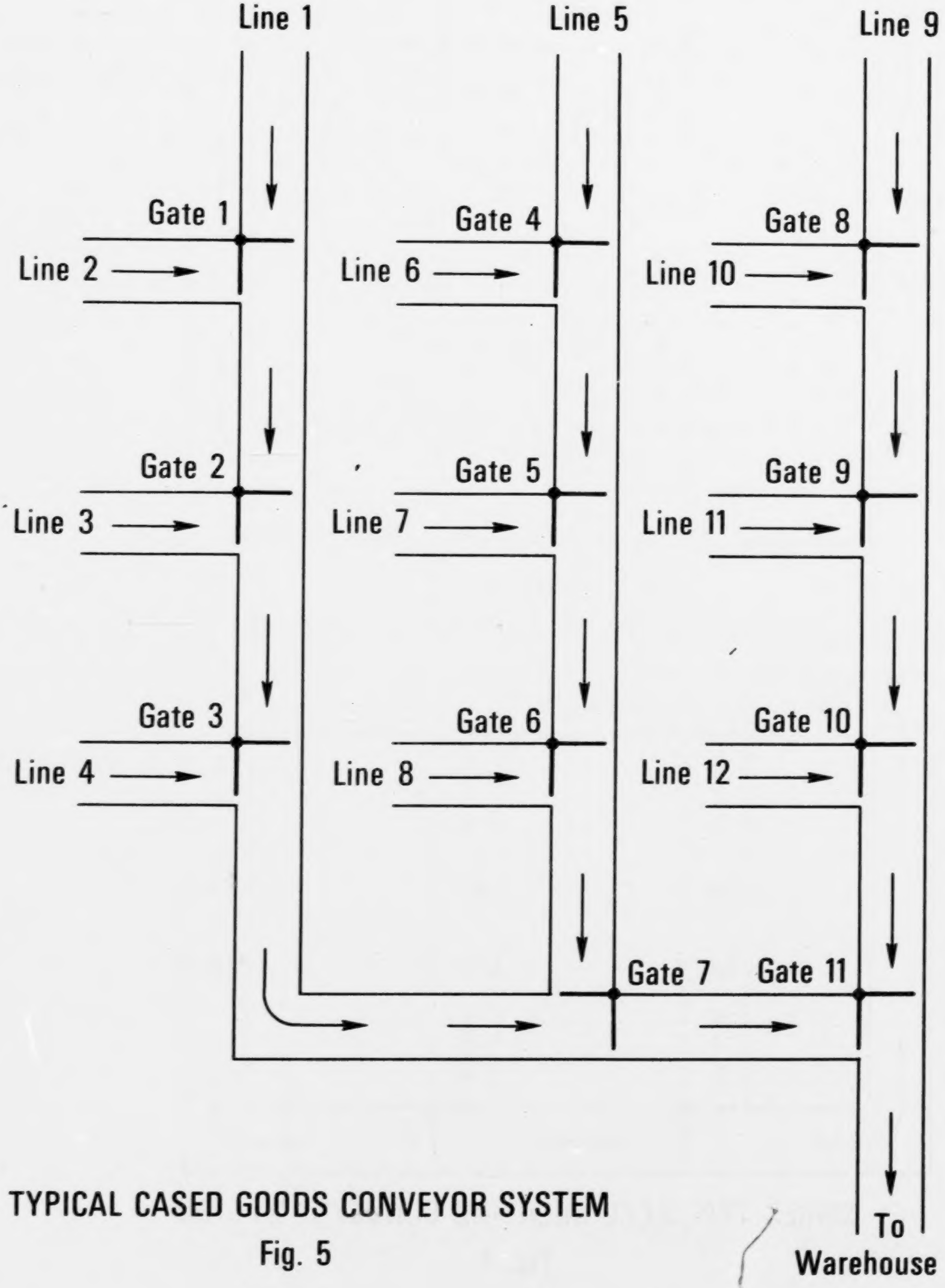
CASE B1 ARRIVES
Fig. 2


B
B2
B1
A2
A1
A

CASE B2 ARRIVES, A.ARM SHUTS
Fig. 3

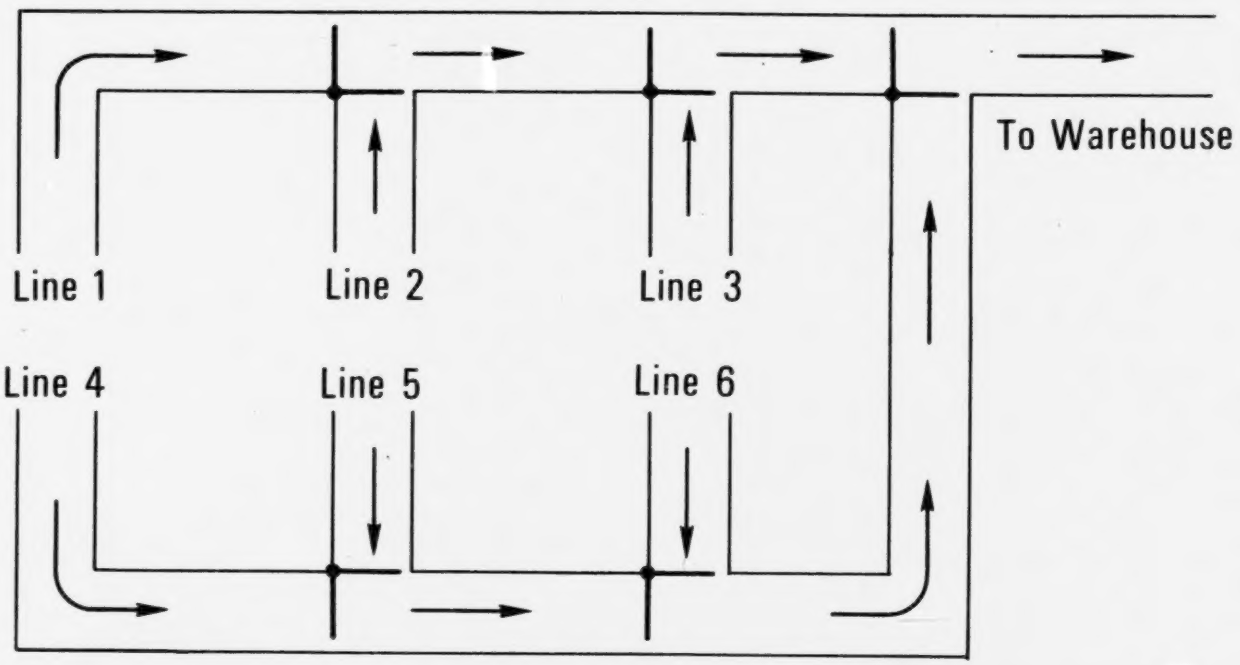


CASE A2 ARRIVES, B.ARM OPENS
Fig. 4


Line 1
Line 5
Line 9
Gate 1
Gate 4
Gate 8
Line 2
Line 6
Line 10
Gate 2
Gate 5
Gate 9
Line 3
Line 7
Line 11
Gate 3
Gate 6
Gate 10
Line 4
Line 8
Line 12
Gate 7
Gate 11
To
Warehouse

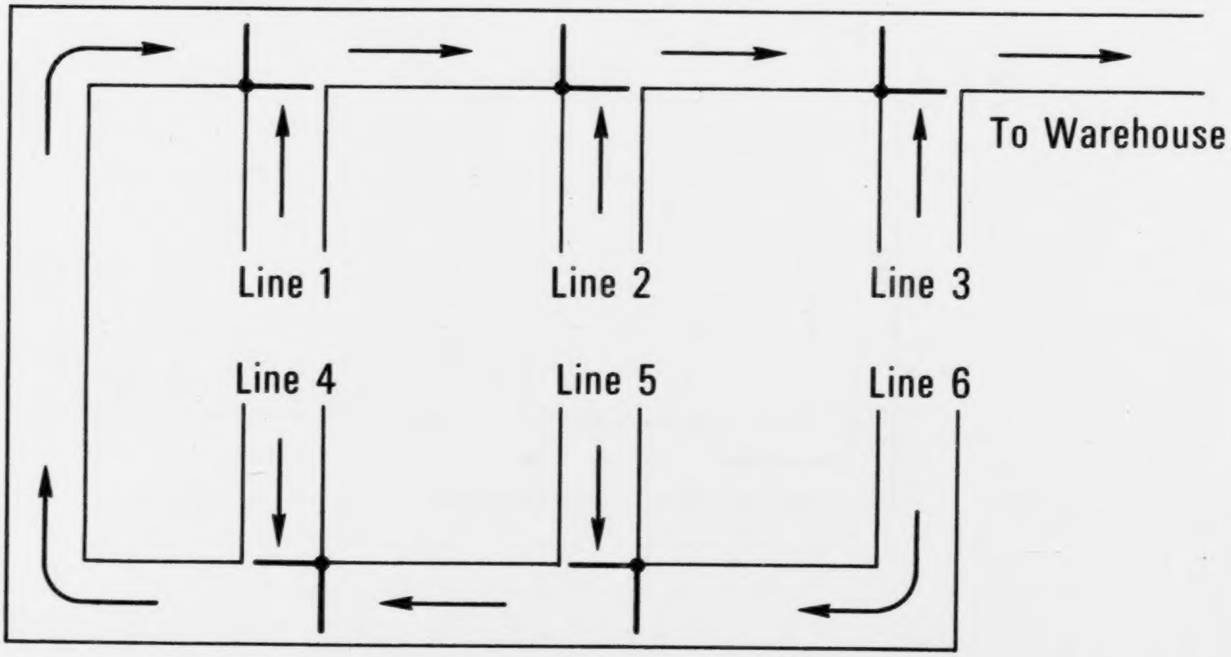
TYPICAL CASED GOODS CONVEYOR SYSTEM

Fig. 5


To Warehouse
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6

PARALLEL-TYPE ACCUMULATING CONVEYOR SYSTEM

Fig. 6


To Warehouse
Line 1
Line 2
Line 3
Line 4
Line 5
Line 6

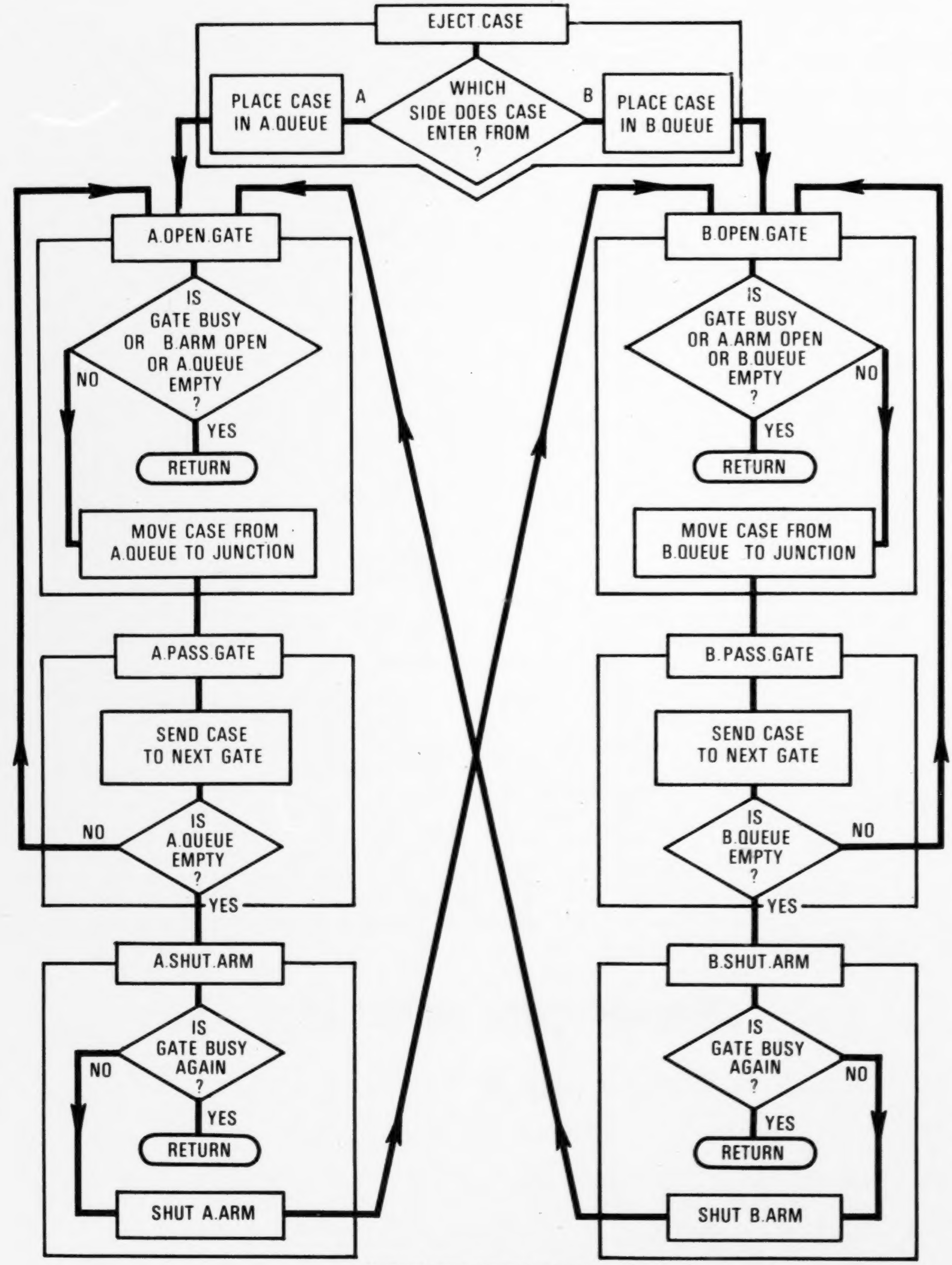
SERIES-TYPE ACCUMULATING CONVEYOR SYSTEM

Fig. 7

```
LINE  CACI SIMSCRIPT II.5 REL/VERS 6C                       07/11/72

      EVENT A.OPEN.GATE GIVEN A.GATE.NUMBER

           ''  THE FOLLOWING "DEFINE TO MEAN" STATEMENTS ARE NORMALLY ''
           ''  FOUND IN THE PREAMBLE.  THEY ARE INSERTED IN THIS ROU- ''
           ''  TINE SO THAT ANY UNEXPLAINED STATEMENTS OR PHRASES WILL''
           ''  BE EASILY UNDERSTOOD.                                  ''
           DEFINE TR2 TO MEAN IF TRACE NE 2, JUMP AHEAD   ELSE
           DEFINE IS.BUSY TO MEAN NE 0
           DEFINE IS.OPEN TO MEAN NE 0
           DEFINE PUT TO MEAN FILE
           DEFINE SECONDS TO MEAN UNITS
           DEFINE BUSY.A TO MEAN 1
           DEFINE ONE TO MEAN 1
           DEFINE EQUAL TO MEAN =

           LET GATE EQUAL A.GATE.NUMBER
           LET TEMP.NO EQUAL A.GATE.NUMBER
      TR2  '' LEVEL 2 TRACE  ''
           PRINT ONE LINE WITH FCT.SECONDS(TIME.V),
                               GATE,
                               STATUS AND
                               B.ARM AS FOLLOWS
  EVENT A.OPEN   **.** SECONDS   GATE **   STATUS = *   BARM = *
      HERE  ''  END OF LEVEL 2 TRACE  ''
           IF STATUS IS.BUSY
                OR B.ARM IS.OPEN
                     OR A.QUEUE IS EMPTY, RETURN
                          OTHERWISE
                          REMOVE THE FIRST CASE FROM THE A.QUEUE
                          LET INCHES EQUAL LENGTH(CASE)
                          SUBTRACT INCHES FROM A.LENGTH
                          PUT THE CASE IN THE JUNCTION
                          ADD ONE TO COUNTER
                          LET STATUS EQUAL BUSY.A
                          SCHEDULE AN A.PASS.GATE GIVEN TEMP.NO IN
                               PASS.ARM(A.SPEED,INCHES) SECONDS
                               ''  PASS.ARM IS A FUNCTION THAT CALCULATES THE
                               ''  TIME NECESSARY FOR THE CASE TO PASS A GIVEN
                               ''  POINT KNOWING THE SPEED AND LENGTH OF CASE.
      RETURN  ''AND''  END
```

SIMSCRIPT EVENT SOURCE CODE

Fig. 8


EJECT.CASE
WHICH SIDE DOES CASE ENTER FROM ?
A
B
PLACE CASE IN A.QUEUE
PLACE CASE IN B.QUEUE
A.OPEN.GATE
IS GATE BUSY OR B.ARM OPEN OR A.QUEUE EMPTY ?
NO
YES
RETURN
MOVE CASE FROM A.QUEUE TO JUNCTION
A.PASS.GATE
SEND CASE TO NEXT GATE
IS A.QUEUE EMPTY ?
NO
YES
A.SHUT.ARM
IS GATE BUSY AGAIN ?
NO
YES
RETURN
SHUT A.ARM
B.OPEN.GATE
IS GATE BUSY OR A.ARM OPEN OR B.QUEUE EMPTY ?
NO
YES
RETURN
MOVE CASE FROM B.QUEUE TO JUNCTION
B.PASS.GATE
SEND CASE TO NEXT GATE
IS B.QUEUE EMPTY ?
NO
YES
B.SHUT.ARM
IS GATE BUSY AGAIN ?
NO
YES
RETURN
SHUT B.ARM

EVENT RELATIONSHIPS

Fig. 9

```
LENGTH OF SIMULATION
 1 HRS,  0 MINS
SAMPLE RUN FOR SIMCON WITH 6 LINES AND 5 GATES


                               SIMCON CONVEYOR SIMULATOR


LINE INPUT INFORMATION:

 LINE      RATE   CASE LENGTH    FEEDS  SIDE OF
NUMBER  CASES/MIN    INCHES       GATE    GATE
   1      10.00      28.00         1       A
   2       4.00      28.00         1       B
   3       2.50      28.00         2       B
   4       1.00      28.00         3       B
   5       4.00      28.00         4       B
   6       6.00      24.00         5       B



GATE INPUT INFORMATION:

                    -----------SIDE A-----------          -----------SIDE B-----------
 GATE  FEEDS                  SPEED     ARM CLOSING                  SPEED     ARM CLOSING
NUMBER  GATE        SOURCE   PAST ARM      TIME           SOURCE    PAST ARM      TIME
   1      2           L1   100 FT/MIN   2.00 SEC            L2    125 FT/MIN   2.00 SEC
   2      3           G1   100 FT/MIN   2.00 SEC            L3    125 FT/MIN   2.00 SEC
   3      4           G2   100 FT/MIN   2.00 SEC            L4    125 FT/MIN   2.00 SEC
   4      5           G3   110 FT/MIN   2.00 SEC            L5    120 FT/MIN   2.00 SEC
   5     99           G4   110 FT/MIN   2.00 SEC            L6    120 FT/MIN   2.00 SEC



SIMULATION RESULTS:

             MAXIMUM QUEUE              AVERAGE QUEUE
 GATE      LENGTH (IN FEET)           LENGTH (IN FEET)
NUMBER  --SIDE A------SIDE B--  --SIDE A------SIDE B--
   1        2.33        2.33          .17         .16
   2        2.33        2.33          .18         .26
   3        2.33        2.33          .08         .31
   4        4.67        2.33          .55         .45
   5        9.33       14.00          .81        2.32



 GATE     PERCENT        NUMBER OF      PRESENT QUEUE LENGTH
NUMBER  UTILIZATION   CASES PROCESSED    SIDE A    SIDE B
   1       77.27             840         0.  FT    0.  FT
   2       87.95             992         0.  FT    0.  FT
   3       91.34            1053         0.  FT    0.  FT
   4       97.11            1293         0.  FT    0.  FT
   5       99.99            1653         0.  FT    4.0 FT
```

LENGTH OF SIMULATION • SIMCON REPORT

Fig. 10

```
                                      FREQUENCY TABLES

               ---------------------------- UPPER LIMIT,IN FEET ------------------------------------------------ OVER-
                  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20 FLOW

       SIDE A:    0    0  602    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
GATE   1
       SIDE B:    0    0  237    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

       SIDE A:    0    0  840    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
GATE   2
       SIDE B:    0    0  152    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

       SIDE A:    0    0  993    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
GATE   3
       SIDE B:    0    0   60    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

       SIDE A:    0    0  986    0   66    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
GATE   4
       SIDE B:    0    0  241    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0

       SIDE A:    0    0 1112    0  140    0   39    0    0    2    0    0    0    0    0    0    0    0    0    0    0
GATE   5
       SIDE B:    0  196    0   97    0   44    0   19    0    3    0    1    0    1    0    0    0    0    0    0    0
```

FREQUENCY TABLE • SIMCON REPORT

Fig. 11

REFERENCES

1. Kiviat, P. J., R. Villanueva and H. M. Markowitz, The SIMSCRIPT II Programming Language, Prentice-Hall, Inc. Englewood Cliffs, N.J. 1968

2. Stuebing, R. C., "Monte Carlo Simulation of Full Case Conveying Systems", Digest of the Second Conference on Applications of Simulation 1968

A SIMULATOR FOR DESIGNING HIGH-RISE WAREHOUSE SYSTEMS

Kailash M. Bafna, Ph.D.

Assistant Professor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Abstract

This paper presents a simulator to evaluate alternative designs of high-rise warehouse systems using stacker cranes to permit system cost minimization while satisfying design constraints and specifications. There are no warehouse size restraints on its use. Furthermore it can consider single or multiple stacker crane systems. Resolution of problems associated with finding combinations of storage height and aisle length in combination with crane vertical and horizontal travel speeds is provided. Hardware and operating variables considered are more extensive than those found in earlier simulations of similar systems.

Introduction

Automatic stacker cranes are still in their infancy, being only about a decade old. However, in this short period of time, they have assumed an increasingly important role in the field of materials handling, especially in warehousing. It is predicted that the rate of growth of this industry over the next few years will approach an exponential rate and form a significant proportion of the materials handling industry.

Stacker installations involve high investment. In 1965, the average system cost $150,000; today it costs over $1 million; in 1975, it may range from $4 to $5 million.[1] At the time management makes a commitment for a stacker installation (which may include new racks in an existing warehouse, or even building a new warehouse), risks are involved in forecasting operating demands and capacities and it is not absolutely certain whether the desired rate of return on investment will be achieved.

Management makes an investment for which it expects a certain return to be realized, and would like to minimize their risk in realizing such returns. Such a preview can only be attained if the engineers (who may have no prior experience with such an installation) have simple analytical and quantitative tools available by which they can study the characteristics of alternative systems. Such techniques may also be used for estimating realizable values of the principal parameters of a high-rise warehouse using stacker cranes prior to final design and installation.

---

[1] "Up-to-Date Report: How Stacker Cranes Stack Up", Modern Manufacturing, June 1970, p. 56.

This research has approached this objective by developing a discrete-event simulator, hereafter referred to as BASS (Bafna's Stacker Simulator). The simulator has been tested for different throughput requirements.

This paper very briefly describes the cost representation (the cost model for the various components of the warehouse is one of the most elaborate cost representations) of a high-rise warehouse. This is followed by a description of BASS. The paper concludes with a discussion and analysis of some sample runs using the simulator.

Principal Assumptions

Some of the major assumptions made in the analysis are:

1. The physical size, maximum height and weight of the unit-load to be stored in the system is known.

2. The system handles only one size of unit-loads.

3. An estimate of the total storage slot requirements in the warehouse is known.

4. The required throughput for the system (number of unit-loads to be handled--both stored and retrieved--by the proposed system) is known.

The Cost Model

A high-rise warehouse using stacker cranes consists of several elements such as land, building, racks, stacker cranes, transfer cars, fire protection equipment, peripheral equipment (such as conveyors, fork lift trucks, etc.), inventory control equipment, lighting and heating (if necessary), etc., a cost being associated with each. The costs for a few of these elements will remain somewhat constant irrespective of the warehouse design. However, the costs of the other elements will vary with the warehouse and stacker crane parameters. Hence, in order to select between alternate warehouse designs, it will suffice to study the costs that vary with variations in the warehouse and crane parameters.

The design variables used in the analysis are:

1. Number of slots along the height of rack.

2. Pairs of rows of racks in the system (also equal to the number of aisles).

3. Number of stacker cranes in the system.

4. Horizontal speed of stacker cranes.

5. Vertical speed of stacker cranes.

The variable costs which have been considered in the cost model are described below.

1. Cost of Floor Space. This is considered to be the cost of (a) area occupied by

racks, (b) area occupied by aisles, and (c) area for related warehouse services.

2. Cost of Building. This is the sum of three elemental costs:

   1. Foundation and floor on which the racks are mounted.
   2. Roof of warehouse.
   3. Perimeter walls.

The costs of foundations and floor vary with the intensity of loading (the height of storage comes into play here). The cost of roof and the perimeter walls varies with the roof height.

3. Cost of Racks. The cost of racks is treated as the sum of four elements:

   1. Cost of material and labor for load arms and ties.
   2. Cost of material and labor for the pairs of columns required for each truss assembly. Changes in the cross-sectional area of the columns due to increased loadings when the rack height is increased have been considered here.
   3. Cost of splicing if there are any splices.
   4. Cost of installing the racks.

4. Cost of Stacker Cranes. The cost of each crane is analyzed as the sum of four elements:

   1. Cost of stacker crane hardware which consists of the stacker base, the vertical mast and the lifting mechanism hardware (elevator). It provides for the variation in the cost of the mast due to increasing heights.
   2. Cost of providing a specific horizontal speed.
   3. Cost of providing a specific vertical speed.
   4. Cost of the controls. Each stacker crane design will have its own control design. In addition, the control of several individual stacker cranes may be combined for central control. The cost of operators required depending upon the level of controls is also considered.

5. Cost of Transfer Cars. This includes the cost of all the transfer cars to transfer the cranes for the given size and load requirements.

6. Cost of Fire Protection. The protection against fire appears to be best accomplished by a well designed system of automatic sprinklers, or a combination of sprinklers with high expansion foam. It may also be preferable specifically to have one of these types due to the kind of material to be stored or other constraints. Both types have been considered in the analysis and it is possible to select either one or the cheaper of the two.

Annual Cost of the System

Having determined the individual elements of cost, the equivalent annual cost can be computed. The salvage values and the write-off periods for the various elements of the system are treated as variables.

The Simulator

The operation of a high-rise warehouse using stacker cranes has been programmed into the simulator, BASS. Since warehouses vary in their layouts, requirements, and operating rules, a basic layout of a high-rise warehouse and operating rules in a general warehouse have been modelled. If any deviations from these are required, it can be done by making changes in the corresponding subroutines of the simulator.

Each aisle in the warehouse is assumed to have two queues--a material queue and an order queue. In addition, there is a marshalling area where a "common material queue" is formed. Each material queue has a finite capacity. When all of these are full, the overflow is sent into the common material queue (unassigned materials). Flow of material from the common queue to the material queue of a specific aisle takes place automatically as soon as a vacancy occurs.

Arrival of material to be stored and orders to be retrieved from the system are input based upon user demands, or alternatively, according to any desired distribution. When more than one aisle is serviced by a stacker crane, operating decisions for transfer between aisles are made by a choice of criteria including maximum waiting time before servicing orders, maximum number of orders allowed in a queue, completion of a given number of cycles in an aisle, and the material and order queues for the aisles becoming empty. BASS keeps track of the full or empty condition of each storage location and reacts to the slot condition in arriving at storage or retrieval decisions.

The simulator has the capability of evaluating alternative operating policies related to scheduling storage and/or retrieval cycles to meet operating schedules. During the simulation run, the stacker crane performs single address (deposit only or retrieve only) or dual address (deposit and retrieve) cycles as necessary depending on the available entries in the material and order queues.

Language Used

The language used in BASS is Fortran IV. To facilitate the programming of standard simulation procedures, GASP II, a Fortran IV based simulation language is used. The purpose of selecting Fortran IV as a programming language for BASS is twofold. First, Fortran compilers are commonly available. Second, BASS may need minor changes of decision-rules, etc., to suit specific needs. Fortran makes this relatively easy since a high percentage of programmers are familiar with Fortran who, with some knowledge of GASP (an event-oriented simulation language), can make the necessary changes. Besides, greater portability in running the simulator has been achieved by having the maximum length of variable names limited to five letters. This allows BASS to be run on a variety of computers.

Design Variables

The design variables described above have been used in BASS. The names assigned to them are:

KHGHT -- number of slots along height.

NAILE -- number of aisles.

NSCR -- number of stacker cranes.

HOVEL -- horizontal speed of stacker.

VEVEL -- vertical speed of stacker.

Since the first three variables can be incremented in steps of one, the upper and lower limits to be considered by the designer are specified for each of these. All possible values of the two speeds are also specified.

Operation of BASS

The principal stages of BASS are shown in the system flow chart in Figure 1. The lower limits (as specified by the programmer) of KHGHT, NAILE, NSCR, HOVEL, and VEVEL as fed in by the data cards are taken as the initial system variables. The throughput is simulated with this set of variables until the steady-state conditions, as verified by programming procedures, have been reached. This calculated value of throughput is transferred to the MAIN program and compared with the last throughput (initially set as 0). If it has increased, a new series of variable value sets are generated (this is done by increasing in each set one of the five variables to its next possible higher value) and the set giving the lowest annual cost is transferred to GASP to simulate the next throughput with. If the throughput has not increased, another set of variables (the one giving the next higher annual cost) from the series generated earlier is transferred to GASP and the throughput is calculated. This cycle is repeated over and over again until the calculated throughput is greater than the required throughput. This throughput is then checked (using the same set of variables) under more stringent steady-state conditions than before. If the revised value of the throughput still equals or exceeds the required throughput, the simulation stops and the final values of the design variables are printed. If the revised throughput is less than required, the procedure is repeated until the final requirements are reached.

Outputs from BASS

For each set of variables, BASS collects statistics on waiting times, cycle lengths, times between transfers, crane utilization, times lost in transferring and travelling empty, material and order queue build-ups in each aisle, number of dual and single address cycles by each crane, and the throughput for the system. In addition, it provides histograms on waiting times and cycle lengths. It also computes the costs and the throughput for each set of variables.

Special Features of BASS

Special features built into the simulator, to suit the specific needs of individuals who may use it, include:

1. Any size warehouse can be simulated, the only limitation being available core storage in the computer system.

2. Depending on individual requirements, the start up of the stacker crane at the beginning of simulation may be handled in either of two ways. First, the cranes can be started

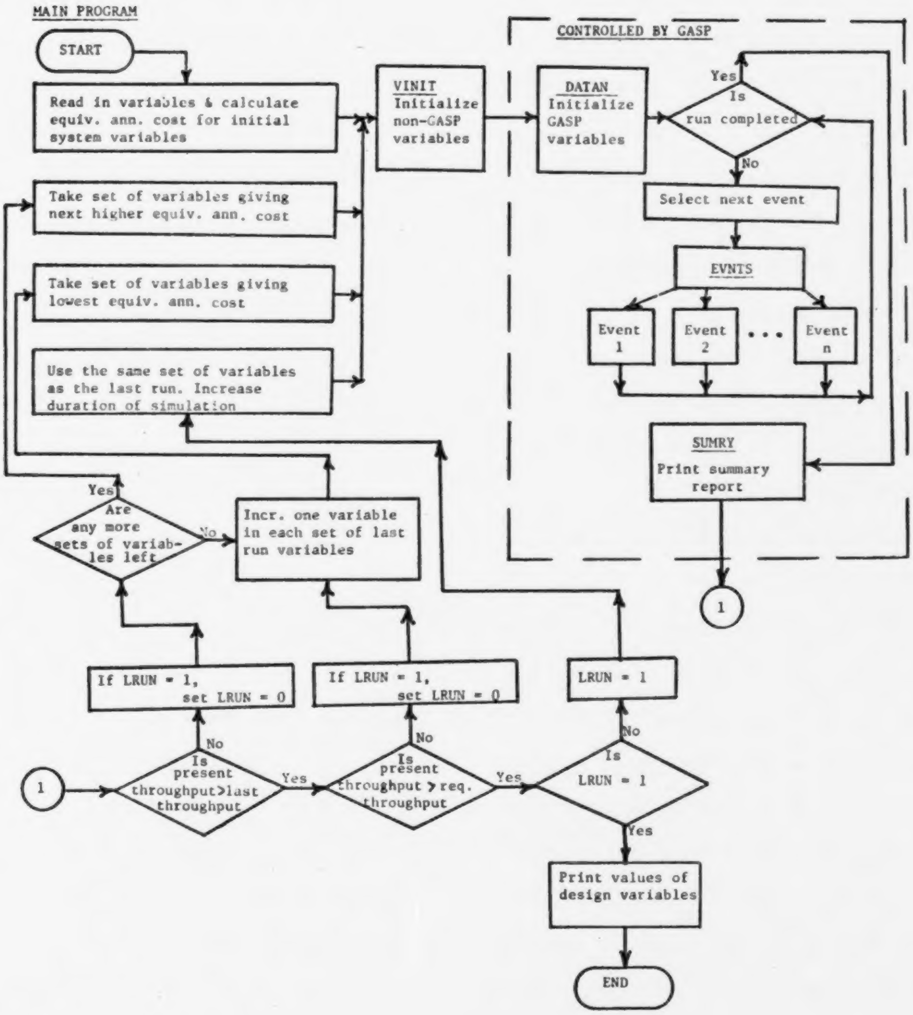

MAIN PROGRAM
START
Read in variables & calculate equiv. ann. cost for initial system variables
Take set of variables giving next higher equiv. ann. cost
Take set of variables giving lowest equiv. ann. cost
Use the same set of variables as the last run. Increase duration of simulation
VINIT Initialize non-GASP variables
CONTROLLED BY GASP
DATAN Initialize GASP variables
Is run completed
Yes
No
Select next event
EVNTS
Event 1
Event 2
Event n
SUMRY Print summary report
1
Are any more sets of variables left
Yes
No
Incr. one variable in each set of last run variables
If LRUN = 1, set LRUN = 0
If LRUN = 1, set LRUN = 0
LRUN = 1
1
Is present throughput>last throughput
Yes
No
Is present throughput > req. throughput
Yes
No
Is LRUN = 1
No
Yes
Print values of design variables
END

Figure 1. System flow chart showing the principal functions performed by the main routines.

whenever entries are available in the queues, i.e., the first arrivals can trigger the start of the cranes. Alternately, the cranes can start at any desired and predetermined time. All arrivals until that time are kept in queues. All that the user has to do is to specify start times of stacker cranes, such as start of shift, one-quarter hour after shift start, etc.

3. Also, the queues can be emptied at the end of the simulation before statistics are computed, or alternately, queues can be left in their end of simulation conditions and statistics collected without runout. This feature is available by assigning a value of 0.0 or 1.0 to a variable, QUEUE.

4. Since the value of QUEUE is checked (in

either case) at the end of the simulation, other analysis to occur at the end of simulation can be programmed into BASS, to be called when QUEUE has specific values. This will involve changing of a few cards in the end of cycle subroutine. This feature provides greater flexibility to users to add on special requirements without extensive program changes.

5. Depending on the amount of money or computation time available to the user, he can adjust the duration of each run to meet his specific needs of accuracy.

6. Flexibility is provided to handle situations where users do not have an accurate idea of the ranges in which the number of aisles, height of racks, and the number of cranes fall. The simulation can be done using 1 aisle, 1 crane, and 1 slot height (although this is an infeasible value, it is used to emphasize the point) as the starting point and working up until the required solution is reached. Of course, the user will have to pay for the lack of the required knowledge in increased processing time.

7. By setting the value of a variable LFIRE as 1, 2, or 3, it is possible to select the system having fire protection with sprinklers only, foam only, or the one with least cost.

8. The arrivals of materials to be deposited in the warehouse and the orders to be retrieved from the warehouse can be made to follow any given distribution with a minimum of change in the program. In addition, empirical data collected from an existing warehouse can also be used to generate the arrivals.

9. Great flexibility is provided by the fact that Fortran IV had been used as the programming language. This provides for ease in programming new decision-rules and events to suit individual requirements.

From the aforegoing, it can be concluded that BASS is a flexible simulator. The need for this flexibility arises from the fact that high-rise warehouses using stacker cranes are still in their infancy and numerous developments and changes are anticipated in the future. Because of the nature of BASS, these changes can be programmed into it without having to develop an altogether new simulator.

Validating the Simulator

In order to validate BASS, several runs have been made with different storage requirements, each with varying throughput requirements. The storage requirements selected were 600 slots, 5,900 slots, 13,500 slots, and 14,500 slots. The simulator ran for all of these sizes indicating that it could be used to design very small systems as well as large ones.

The results obtained at each step in one of these runs are summarized in Table 1. The run was made for a warehouse having approximately 5,900 storage slots and a throughput capacity of 90/hour. Exponential arrivals of materials for depositing and orders for retrieval were assumed. The values of the variables for each iteration

Table 1

Step-by-step results of a simulation run. Total storage required = 5,900 slots (approx.), and hourly throughput required = 90.0.

ITERATION	NO. OF AISLES (NAILE)	HEIGHT IN SLOTS (KHGHT)	LENGTH OF RACKS IN SLOTS (KLGTH)	NO. OF STACKER CRANES (NSCR)	HORIZ. SPEED (FT/MIN) (HOVEL)	VERT. SPEED (FT/MIN) (VEVEL)	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGH-PUT (PER HR) (HTHPT)
1	3	5	197	1	262	74	240,272	71.04
2	4	5	148	1	262	74	228,977	73.02
3	5	5	118	1	262	74	222,781	74.28
4	6	5	99	1	262	74	221,404	74.70
5	6	5	99	1	420	74	221,609	73.14
6	6	5	99	1	262	99	221,650	74.46
7	7	5	85	1	262	74	222,484	74.70
8	6	6	82	1	262	74	238,630	75.78
9	6	7	71	1	262	74	232,137	77.40
10	6	8	62	1	262	74	227,876	78.78
11	6	9	55	1	262	74	225,247	80.22
12	6	10	50	1	262	74	225,185	81.96
13	6	11	45	1	262	74	223,659	83.16
14	6	12	41	1	262	74	223,786	84.18
15	6	12	41	1	420	74	223,991	82.98
16	6	12	41	1	262	99	224,033	83.40
17	6	13	38	1	262	74	225,081	84.66
18	6	13	38	1	420	74	225,286	83.52
19	6	13	38	1	262	99	225,327	84.24
20	6	14	36	1	262	74	228,804	84.06
21	7	13	33	1	262	74	231,114	84.36
22	6	13	38	2	262	74	242,900	97.32
23	4	5	148	2	262	74	246,386	95.70
24	3	5	197	2	420	74	252,957	100.48

are shown in the table. It also shows the equivalent annual cost and the hourly throughput obtainable with the system for each run.

The results of the above run are plotted in Figure 2, which shows how, at each step, the simulator tries to increase throughput while minimizing the incremental cost. The plot has a zig-zag formation. Up to iteration 4, cost decreases with increases in throughput. Iteration 5, 6, and 7 do not show any increase in throughput. Finally, there is an increase in throughput in 8, but with a significant increase in cost. Once again cost generally keeps decreasing for increases in throughput up to iteration 17. Subsequent iterations fail to increase throughput and hence, again in iteration 22, the cost increases significantly for an increase in throughput.

Since the increased throughput in 22 is obtained by increasing NSCR from 1 to 2, the initial values of the variables are used with NSCR = 2 for 23 (Table 1). Since the throughput of 22 is better than that of 23, hence run 24 is made which yields a better throughput. However, EQACT of 24 is greater than that of 22 and hence iteration 22 is the final design.

Sensitivity Analysis

BASS is intended to aid the designer in his process of decision-making. The sensitivity of the cost to changes in the throughput at each iteration could be a valuable guide in the decision-making.

Table 2 shows the values of cost and throughput at each iteration for the data given in Table 1. The fourth column is the equivalent annual cost per unit increase in throughput at each iteration. This is plotted in Figure 3. Since this plot is for a low throughput system, it is found that the incremental cost per unit throughput is lower for higher throughputs. In iteration 22, the number of stacker cranes is increased from 1 to 2. This gives a sharp increase in throughput (about 13 units) for an increase in cost of approximately $18,000. Since a through-

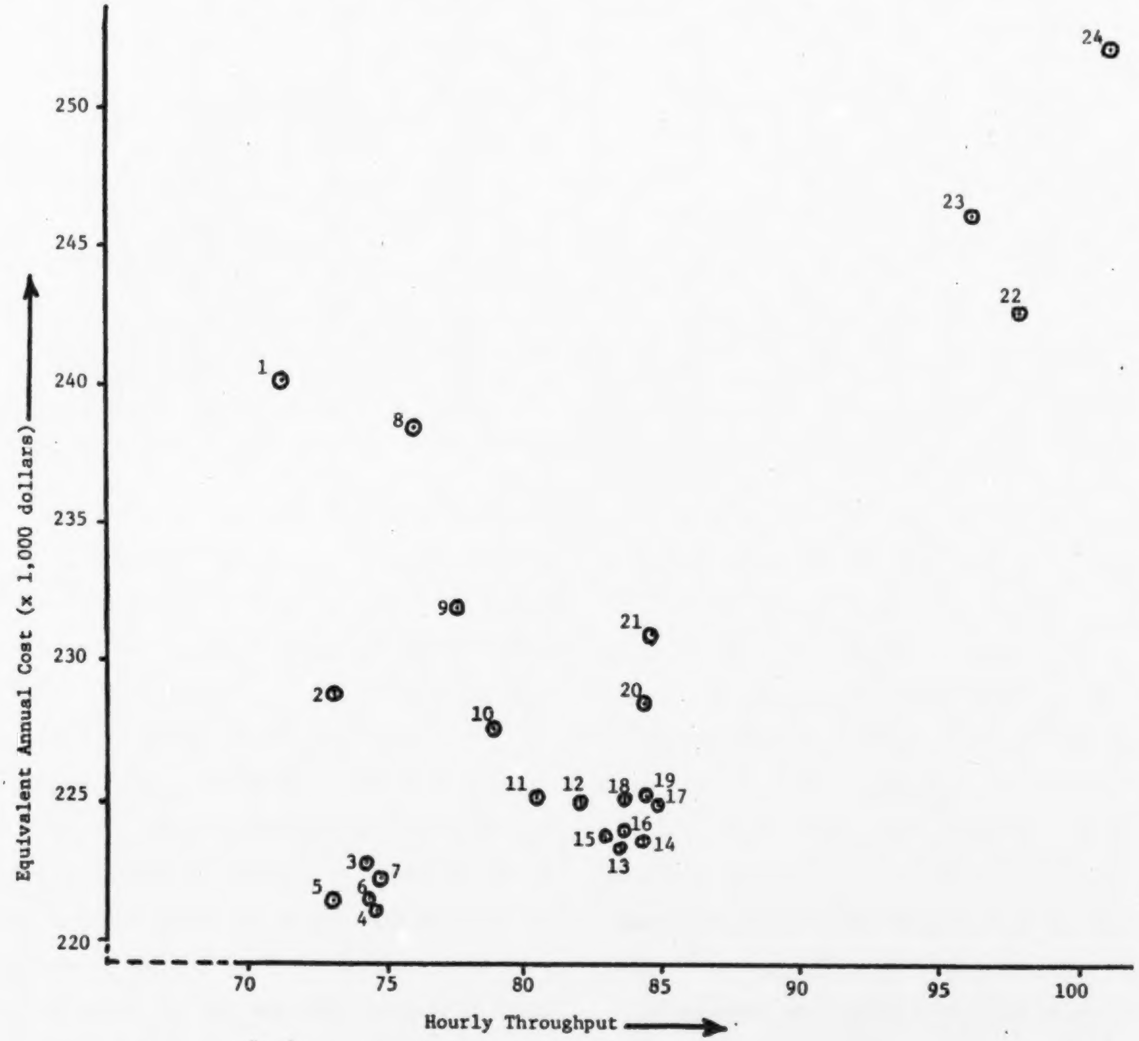


Figure 2. Plot of equivalent annual cost vs. throughput for each iteration.

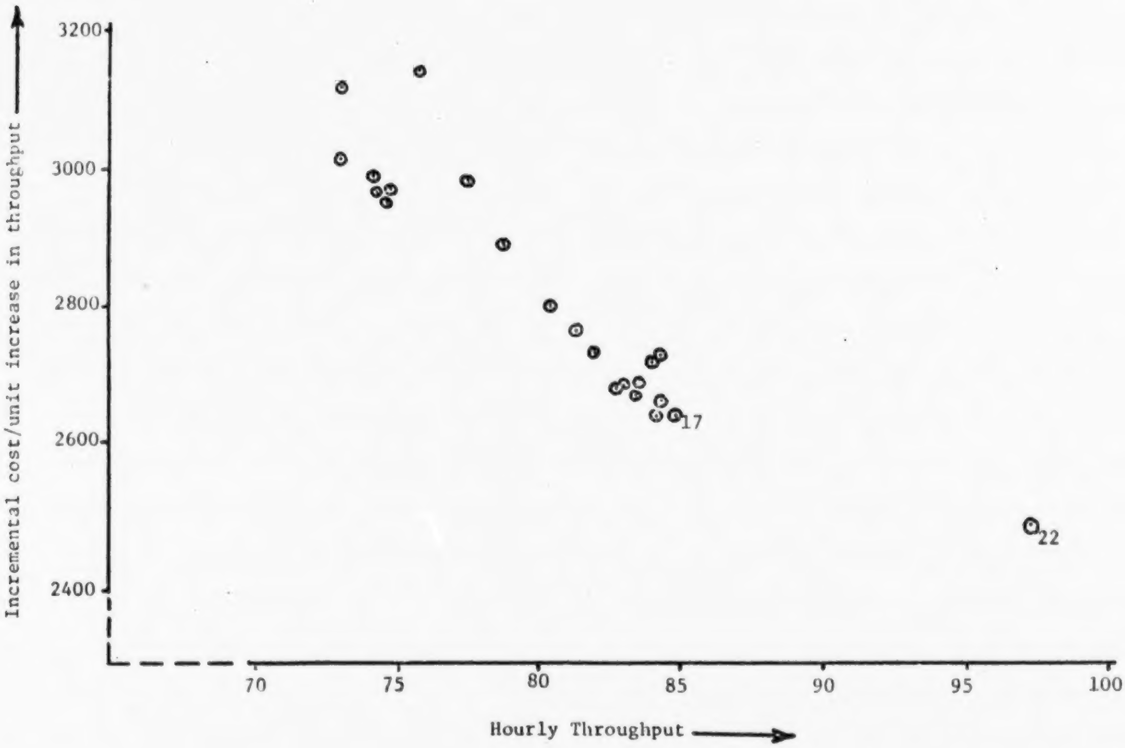


Figure 3. Incremental equivalent annual cost per unit increase in the throughput at each iteration.

Table 2

Values of equivalent annual cost and throughput at each iteration.

ITERATION	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGHPUT (PER HR.) (HTHPT)	EQACT / HTHPT	ITERATION	EQUIV. ANNUAL COST ($) (EQACT)	ACTUAL THROUGHPUT (PER HR.) (HTHPT)	EQACT / HTHPT
1	240,272	71.04	3,382	12	225,185	81.96	2,747
2	228,977	73.02	3,136	13	223,659	83.16	2,690
3	222,781	74.28	2,999	14	223,786	84.18	2,658
4	221,404	74.70	2,964	15	223,991	82.98	2,699
5	221,609	73.14	3,030	16	224,033	83.40	2,686
6	221,650	74.46	2,977	17	225,081	84.66	2,659
7	222,484	74.70	2,978	18	225,286	83.52	2,697
8	238,630	75.78	3,149	19	225,327	84.24	2,675
9	232,137	77.40	2,999	20	228,804	84.06	2,722
10	227,876	78.78	2,893	21	231,114	84.36	2,740
11	225,247	80.22	2,808	22	242,900	97.32	2,496

put rate of about 90 per hour is desired, selection could be on either side of it. The lowest cost per unit throughput for a throughput below 90 was in iteration 17. The value above 90 was in iteration 22. Because of the large difference in the incremental cost at these two points ($163 per unit throughput), it would be better to select the results of iteration 22. However, had the two points been fairly close, either could have been selected, the decision then being based upon how much capital could be made available for the warehouse.

In addition to making a decision on the basis of the sensitivity analysis, information printed out in the summary reports should be analyzed. More specifically, information such as statistics of material and order waiting times, crane utilization, statistics of aisle material queues, order queues, the common material queue, and the types of crane cycles should be studied. The waiting time, especially that of servicing orders, should not exceed that required by the system that the warehouse is to service.

Findings From Runs

The following are some of the main findings from the results of the sample runs made to test the simulator:

1. Whereas it may be necessary from an efficiency standpoint to have the ratio between the horizontal and vertical speeds conform to the ratio between the length and the height of racks, it is not absolutely necessary to have this relationship for a least cost solution.

2. The general belief among stacker manufacturers that higher warehouses are cheaper is validated by the results obtained from the sample runs.

3. Greater travel speeds are not necessary for low throughput systems and basically just add to the cost.

4. For a given system throughput, it is found that the larger the storage capacity of the warehouse, the lower is the cost per storage slot. Also, the cost per unit slot increases as greater throughputs are desired.

Conclusion

The results from the sample runs show that numerous alternatives confront the designer of high-rise warehouse and stacker crane systems and these differ from one another in small steps. It is, therefore, justifiable to conclude that the use of BASS, along with the sensitivity and report analysis described, will prove to be a very useful tool. Besides helping in design selection, it will provide useful information about the actual system which has been either impossible or impractical by earlier analysis techniques. The simulator can also serve as a tool to test different operating rules and management policies related to high-rise storage.

Tutorial 1: Fundamental GPSS Tutorial
Chairman: Thomas J. Schriber, University of Michigan

Approach to modeling in GPSS. Fundamental GPSS blocks, including GENERATE, TERMINATE, SIEZE and RELEASE, ADVANCE, and QUEUE and DEPART. A GPSS "case study" model for a one-line, one-server queuing system. Internal logic of the GPSS processor, including the current and future events chains, and a numeric example explaining how the processor simulates with the one-line, one-server model.

Tutorial 1 Continued: Intermediate GPSS Tutorial
Chairman: Thomas J. Schriber, University of Michigan

Continuation of Wednesday afternoon's fundamental GPSS tutorial. Descritpion of unconditional-mode TRANSFER block, CLEAR card, the ENTER and LEAVE blocks, the RESET card, and the statistical-mode TRANSFER block, with 5 "case studies" illustrating their applications. Consideration of 2 "advanced" GPSS case studies, with rapid overview of the GPSS capabilities they use.

The last 40 minutes of this session will be devoted to presentation of the proceedings paper entitled "One the Application of User Chains in GPSS". This paper is intended for active users of the language, and should also be of value to those whose knowledge of GPSS is limited to that gained in the two tutorials.

ON THE APPLICATION OF USER CHAINS IN GPSS

THOMAS J. SCHRIBER
GRADUATE SCHOOL OF BUSINESS
THE UNIVERSITY OF MICHIGAN
ANN ARBOR, MICHIGAN

ABSTRACT

The GPSS Processor uses a Current Events Chain, Future Events Chain, Interrupt Chains, and Matching Chains to support the logic of a GPSS simulation. These chains, which are an implicit part of the language, are automatically maintained and manipulated by the Processor as a simulation proceeds. At the analyst's option, one or more additional chains of a type known as User Chains can be explicitly incorporated into a GPSS model. These user-defined chains can be introduced for either one of two quite distinct reasons: (1) to decrease the execution time requirements of a given model, and (2) to implement queue disciplines other than first-come, first-served, within Priority Class. Despite their application scope, however, there are several subtleties associated with User Chain use. These subtleties arise principally because the GPSS Processor is inherently sequential in nature. This paper, presented in the spirit of a tutorial, explores User Chain applications and identifies some of the subtleties associated with their use.

1. Introduction

Those to whom this paper is addressed are assumed (a) to be active GPSS model-builders, thoroughly conversant with the operation of the Current and Future Events Chains in the language, but (b) without prior knowledge of the GPSS User Chain entity. The first assumption makes it possible to avoid starting the paper at too elementary a level. The second assumption provides an excuse to include here the fundamental User Chain groundwork needed to support some of the points to be made. Apart from these conveniences, are the assumptions realistic? The evidence suggests that they are. In much GPSS modeling, it is not necessary to apply User Chains (even though their application might be of advantage in decreasing the CPU time required for a simulation). Furthermore, the topic of User Chains is an "advanced" one in GPSS. The self-taught GPSS model-builder, then, can conveniently avoid getting into the User Chain concept. And the person who has "gone through a course" on GPSS may not have been told much, if anything, about User Chains, unless the course was either "long" or "advanced". Finally, there is no definitive treatment of User Chains anywhere in the literature on GPSS. They are introduced, yes, and the mechanics of using the two GPSS "Blocks" associated with them are spelled out, but there are few examples given for them. It is not unusual to see only one or two examples showing how User Chains can be applied when a constrained resource is being simulated with a single GPSS Facility. But this is the simplest, most straightforward application of User Chains. As such, it gives no hint of the subtle "simultaneity of events" complications which are associated with modestly more imaginative User Chain use.

There seems to be a need, then, for a more complete treatment of the GPSS User Chain entity. An attempt is made here to provide that treatment. In total, 7 different Block Diagrams, or Block Diagram segments, are presented and discussed to illustrate User Chain use in various ways [a]. Particular emphasis is given to the "simultaneity of events" problems that can occur in conjunction with User Chains. After the examples presented here have been studied, the GPSS model-builder should be able to apply User Chains creatively, and properly, in whatever contexts might be encountered in practice.

---

[a] Portions of this material are taken from the manuscript for a book being written by Thomas J. Schriber (see reference [1]). As part of the manuscript, these portions have been copyrighted by Professor Schriber, and are reproduced here with his permission.

2. The Concept and Utility of User Chains

Whenever a Transaction encounters a blocking condition during a simulation, it is left, by default, on the Current Events Chain by the GPSS Processor. There are two disadvantages associated with this default Processor behavior.

(1) The CPU time required to simulate with the model may be larger than necessary. This is true even though the Processor makes a distinction between "unique", and "non-unique", blocking conditions in a model. The distinction is made because certain CPU time economies can be realized through the "Scan Indicator" concept whenever a blocking condition is unique. Transactions experiencing unique blocking are "scan-inactive" [b]. Even scan-inactive Transactions are processed at least one time, however, at each reading of the simulation clock. It is true that the only CPU time used to process scan-inactive Transactions is that required to test their Scan Indicators. In the long run, however, even this CPU time can be significant. Furthermore, when blocked Transactions are scan-active, the Processor attempts to move them into their next Block each time they are encountered in the scan, even though the logic of a given situation may make it evident (to the analyst, not to the Processor) that a blocking condition is still in effect. It should be clear, then, that if blocked Transactions could be made totally inactive in a model by removing them from the Current Events Chain, execution time economies could result.

(2) The second potential disadvantage concerns queue discipline. The ordering of blocked Transactions on the Current Events Chain is determined solely by their Priority Level, and the chronological sequence in which they were hooked onto that chain. This is why the default queue discipline in GPSS is "first-come, first-served, within Priority Class". If some other queue discipline were to be implemented, these steps would have to be performed.

(a) Instead of leaving waiting Transactions on the Current Events Chain, they would have to be removed from that chain and put "someplace else".

(b) Then, when the time came for one of them to move forward in the model (to capture a now-available server, for example), the Transaction brought from that "someplace else" and put back on the Current Events Chain could be selected by some criterion other than "first-come, first-served, within Priority Class".

In summary, there are two possible benefits to be realized if blocked Transactions can be removed temporarily from the Current Events Chain. The time required to simulate with a model can conceivably be decreased; and arbitrarily-defined queue disciplines can be implemented.

For the reasons cited, an entity known as "User Chains" has been made a part of the GPSS language. User Chains are a "someplace else" where Transactions can be when they are in a model, but are not on the Current Events Chain or one of the other "implicit" chains. Like the Current and Future Events Chains, User Chains have a "front" and a "back". But here the similarity stops. In the case of the Current and Future Events Chains, the GPSS Processor automatically moves Transactions to and from them, and maintains a pre-defined ordering property for Transactions on them. In the case of User Chains, Transactions are hooked onto them only according to logic explicitly provided by the analyst. Furthermore, the analyst can choose from several available alternatives to determine the position a Transaction is to occupy on a User Chain when it is placed there. In like fashion, Transactions are unlinked from User Chains and brought back into active status only according to the analyst's explicitly-provided logic. The analyst can also choose from a series of options in selecting the one or more Transactions which are to be unhooked from a User Chain and put back onto the Current Events Chain.

This overall logic of User Chain use is shown schematically in Figure 1. Blocks in the figure have been labeled A, B, C, D, E, and F. Blocks C, D, and E suggest the basic sequence followed to simulate use of a limited resource, such as that modeled with a Facility or Storage. C, D, and E might be a SEIZE-ADVANCE-RELEASE combination, or an ENTER-ADVANCE-LEAVE sequence. Block A represents the "look-ahead" feature of User-Chain logic, and block B indicates the consequence which follows when the look-ahead reveals that a blocking condition exists. Block F suggests how an active Transaction which has just removed a blocking condition causes a Transaction to be unlinked from a User Chain and brought back to the Current Events Chain, scheduled to make use of the now-available resource.

As might be expected, a pair of complementary GPSS Blocks is used to accomplish the User-Chain logic shown in Figure 1. One of these Blocks corresponds to the "linking logic" shown at A and B in the figure. The other performs the "unlinking logic" shown at F. It is essentially the use of these two Blocks which will be described in this paper.

User Chains have many of the same features as other GPSS entities. There can be many different User Chains in a model. Each chain can be named

---

[b] Familiarity with unique and non-unique blocking conditions and the Scan Indicator is assumed. For an explanation of these concepts, see sections 7.2 and 7.3 in reference [1].

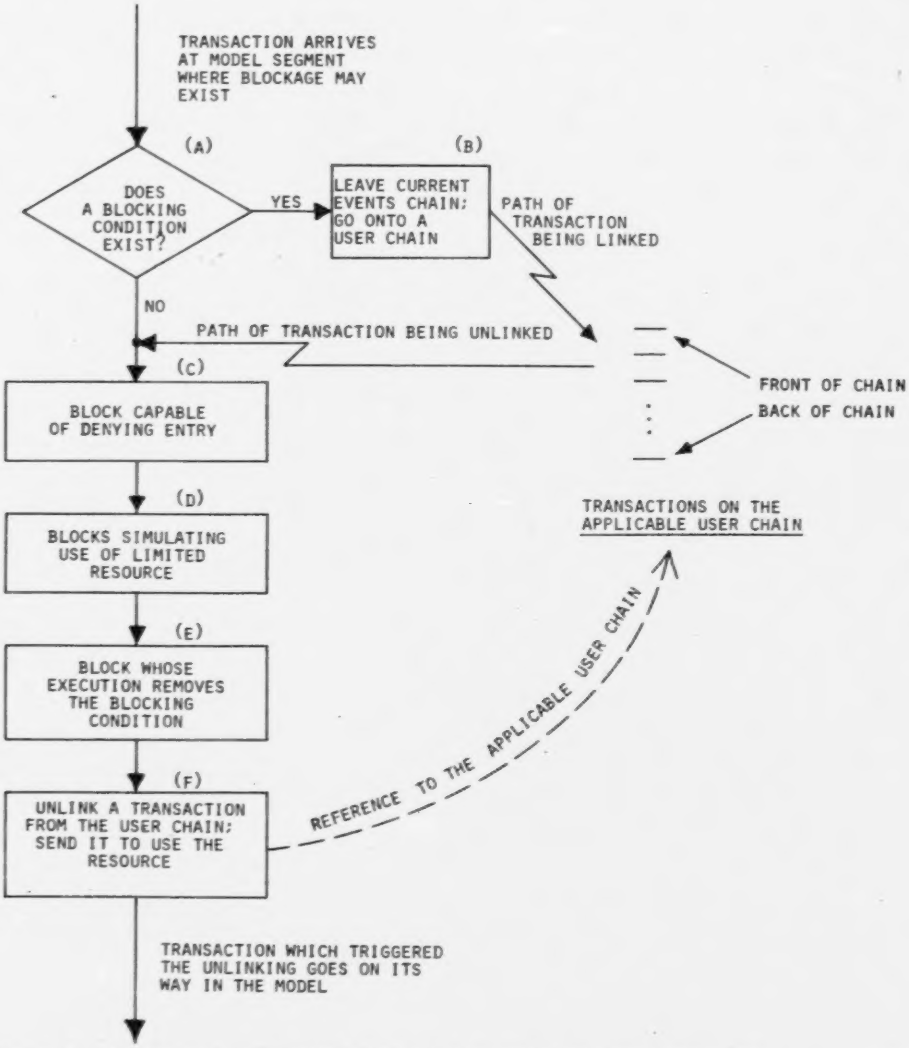

TRANSACTION ARRIVES AT MODEL SEGMENT WHERE BLOCKAGE MAY EXIST
(A)
DOES A BLOCKING CONDITION EXIST?
YES
(B)
LEAVE CURRENT EVENTS CHAIN; GO ONTO A USER CHAIN
PATH OF TRANSACTION BEING LINKED
NO
PATH OF TRANSACTION BEING UNLINKED
(C)
BLOCK CAPABLE OF DENYING ENTRY
FRONT OF CHAIN
BACK OF CHAIN
(D)
BLOCKS SIMULATING USE OF LIMITED RESOURCE
TRANSACTIONS ON THE APPLICABLE USER CHAIN
(E)
BLOCK WHOSE EXECUTION REMOVES THE BLOCKING CONDITION
REFERENCE TO THE APPLICABLE USER CHAIN
(F)
UNLINK A TRANSACTION FROM THE USER CHAIN; SEND IT TO USE THE RESOURCE
TRANSACTION WHICH TRIGGERED THE UNLINKING GOES ON ITS WAY IN THE MODEL

Figure 1 A Schematic Representation of the Logic of User Chain Use

either numerically, or symbolically, according to the usual rules. The number of different User Chains permissible depends on the amount of computer memory available to the Processor. Like Facilities, Storages, Queues, Tables, Blocks, etc., User Chains have a set of Standard Numerical Attributes associated with them. Furthermore, a set of User Chain statistics much like those for Queues appears as part of the standard output produced at the end of a simulation.

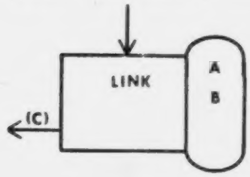
Like the Current and Future Events Chains, nonempty User Chains are printed out by the Processor at the end of a simulation only if "1" is used as the D Operand on the START Card. The PRINT Block can also be used to print out User Chains. For this purpose, the Block's A and B Operands indicate the smallest and largest numbers, respectively, of the User Chains which are to be printed out. The Field C mnemonic is CHA. When a Transaction moves into the Block "PRINT 2,5,CHA", then, User Chains 2 through 5 are printed out as a result.

3. Transaction Movement to and from User Chains: The LINK Block and the UNLINK Block

The ability to put a Transaction onto a User Chain is provided with the LINK Block. The LINK Block can be used in either one of two modes: conditional mode, or unconditional mode. A conditional-mode LINK Block plays the roles of blocks A and B in Figure 1; that is, it embodies a certain "look-ahead" feature, as suggested by block A in Figure 1, and it has the capability of either sending a Transaction to capture an available server, or of putting a Transaction onto a User Chain if there is no available server. In contrast, an unconditional-mode LINK Block has no effective look-ahead capability; it therefore plays only the role of block B in Figure 1. Transactions which enter an unconditional-mode LINK Block are always put onto a User Chain as a consequence.

Because use of the LINK Block in unconditional mode is the easiest to understand, this usage mode will be discussed first. Consider Figure 2, which spells out the specific details associated with the LINK Block. As indicated in that figure, when no C Operand is supplied for the LINK Block, the Block is being used in unconditional-linkage mode. (In fact, if the C Operand were eliminated from the LINK Block in Figure 2, the path leading from the LINK Block would be eliminated, too.) When a Transaction moves into such a LINK Block, it is placed on the User Chain whose name is supplied by the Block's A Operand. The position an incoming Transaction takes on the User Chain is governed by the LINK Block's B Operand. The four-character B Operands FIFO (First-In, First-Out) and LIFO (Last-In, First-Out) cause the Transaction to be placed on the back or front of the chain, respectively. If the B Operand is Pj, where j is some integer from 1 to 100, Transactions are arranged on the User Chain in order of *increasing* Pj value [c]. Each incoming Transaction is placed ahead of those chain residents which have a higher Pj value, but behind those which have a lower Pj value. In case of ties, the incoming Transaction goes behind other residents having the same Pj value. For example, suppose that Transactions A, B, and C have P3 values of -4, 21, and 32, respectively, and they enter the Block "LINK HOLD,P3". Then, after the linking, Transaction A is at the front of the User Chain HOLD, B is behind it, and C is at the back of the chain. If Transaction D now enters the LINK Block and has a P3 value of 21, it is placed between Transactions B and C on the User Chain.

Linking is *conditional* when the LINK Block's C Operand is *used*. A Transaction moving into a conditional-mode LINK Block will either be placed on the User Chain, or will be routed to the "C Block", i.e., the Block in the Location whose name is supplied by the C Operand. In practice, the "C Block" often turns out to be the Block which is sequential to the LINK Block in the model. But even when this is the case, the analyst must use the C Operand on the conditional-mode LINK Block, and must attach the corresponding Location Name to the sequential Block. There is no requirement, however, that the "C Block" be sequential to the LINK Block. This explains why there is a horizontal path leading from the Fig-



Operand	Significance		Default Value or Result
A	Name (numeric or symbolic) of a User Chain		Error
B	Specifies where the Transaction is to be placed on the User Chain; there are three possibilities.		Error
	B Operand	Indication	
	FIFO	Go on the back of the chain	
	LIFO	Go on the front of the chain	
	Pj	Merge into the chain immediately ahead of the Transaction with the next higher value of Parameter j	
C	Optional Operand; Block Location to which the Transaction moves if it is not linked onto the User Chain		Transaction is linked unconditionally onto the User Chain

Figure 2 The LINK Block and Its A, B, and C Operands

ure 2 LINK Block, instead of a vertical path.

---

[c] Some caution is required here. When the LINK Block's B Operand is Pj, the "P" simply signals to the Processor that the linking criterion is "ordered according to the value of a Parameter". The *number* of the Parameter is *directly specified*, and is j itself. If a given LINK Block has P10 as its B Operand, then, the linking criterion is "ordered according to the value of Parameter 10", *not* "ordered according to the Parameter whose number can be found in Parameter 10".

Nothing has been said yet about what determines whether a Transaction entering a conditional-mode LINK Block takes the C-Block exit, or is linked onto the referenced User Chain. To choose between these two possibilities, the GPSS Processor tests the setting of the referenced User Chain's Link Indicator. Each User Chain has its own Link Indicator. The indicator is either "on" ("Set"), or "off" ("Reset"). If the Link Indicator is "off" when a Transaction moves into a conditional mode LINK Block, the Processor does two things.

(1) It turns the Link Indicator "on".
(2) It does not link the Transaction onto the User Chain; instead, it routes the Transaction to the "C Block".

On the other hand, if the Link Indicator already is "on" when a Transaction enters a conditional-mode LINK Block, the Processor puts the Transaction onto the User Chain, and leaves the Link Indicator "on".

As indicated earlier, the unconditional-mode LINK Block corresponds precisely to block B in Figure 2. In this unconditional mode, the referenced User Chain's Link Indicator has no useful purpose. In contrast, the conditional-mode LINK Block takes on the roles played by blocks A and B in Figure 1. The referenced User Chain's Link Indicator embodies the "look-ahead" feature, and can be thought of much in the sense of a green-red traffic light. When the Link Indicator is "off", the traffic light is green. When a Transaction enters a conditional-mode LINK Block and finds that the traffic light is green, it interprets this as a "no blockage" signal. The Transaction moves ahead in the model, but before doing so, it switches the traffic light to red (Link Indicator "on") as a signal for later arrivals to the LINK Block. Conversely, if a Transaction arrives at the LINK Block and finds the traffic light is red (Link Indicator "on"), it interprets this to mean that blockage exists, and consequently goes onto the User Chain instead of moving ahead in the model.

The Link Indicator's look-ahead role cannot be fully appreciated until the Block complementary to the LINK Block has been described, and its effect on the Link Indicator's setting has been indicated. It might be mentioned now, however, that use of the Link Indicator for look-ahead purposes is extremely restricted. In fact, it is really useful as a built-in look-ahead device only when the limited resource which might offer blockage is simulated with a Facility. Most of the time, the analyst supplies his own look-ahead logic with a TEST or GATE Block at position A in Figure 1, and sends Transactions into an unconditional-mode LINK Block when the look-ahead reveals that a blocking condition exists.

The Block complementary to the LINK is the UNLINK. It is the UNLINK Block which is used at position F in Figure 1. The purpose of the UNLINK Block, of course, is to remove one or more Transactions from a User Chain and put them back on the Current Events Chain, so that the Processor can subsequently move them forward again in the model. By using appropriate UNLINK Block Operands, the analyst can specify which Transaction(s) on the User Chain qualify for unlinking. There are two broad possibilities here.

(1) Transactions can be removed from the front or from the back of the User Chain. In this case, Transactions "qualify" for unlinking simply by virtue of the position they occupy on the User Chain.

(2) Transactions can be removed from anywhere on the User Chain, providing that their properties satisfy analyst-specified conditions.

Only the possibilities indicated in (1) above will be described in this section. The possibilities indicated in (2) will be taken up in Section 7.

The UNLINK Block is shown with its various Operands in Figure 3. In considering the Block, it is important to distinguish between the Unlinker-Transaction (i.e., the Transaction which moves into the UNLINK Block, thereby initiating the unlinking operation), and the Unlinkee-Transaction (i.e., the Transactions being unlinked). When a Transaction enters the UNLINK Block, the Processor removes from the referenced User Chain the number of Transactions specified via the C Operand (assuming this many are on the User Chain to begin with, and that they satisfy the unlinking conditions). The C Operand, which can be a constant, a Standard Numerical Attribute, or ALL, is termed the "Unlink Count". If the C Operand is ALL, then all qualifying Transactions on the referenced User Chain will be unlinked. The UNLINK Block's B Operand indicates the Location of the Block to which each of the Unlinked Transactions is to be routed. The D and E Operands are used in combination to indicate from which end of the User Chain the Unlinked Transactions are to be taken. When neither Operand is used, Transactions are unlinked from the front of the User Chain. When BACK is used as the D Operand, and the E Operand is not used, Transactions are unlinked from the back of the User Chain.

The UNLINK Block's F Operand is optional. If it is not used, the Unlinker moves unconditionally from the UNLINK Block to the sequential Block. If used, the F Operand supplies the name of the non-sequential Location to which the Unlinker moves next if no Transactions were unlinked in the attempted unlink operation [d].

---

[d] For the two UNLINK Block D-E combinations in Figure 3, the condition "no Transactions were unlinked can arise if and only if the referenced User Chain is empty prior to the attempted unlinking. For the other UNLINK Block D-E Operand combinations to be discussed in section 7, the "no Transactions were unlinked" condition can occur even when there are Transactions on the User Chain at the time of the attempted unlinking.

Now consider the effect of the UNLINK Block on the referenced User Chain's Link Indicator. When the User Chain is empty at the time a Transaction moves into the UNLINK Block, the Processor switches that User Chain's Link Indicator "off". Using the traffice light analogy, this is equivalent to switching the traffic light from red to green. It is logical for the Unlinker to do this when it has just ceased to cause blockage at a point, and then discovers (because of the empty User Chain) that no other Transaction is currently waiting for the blockage to be removed. Later, when the next Transaction appears at the associated LINK Block, the green traffic light serves as a signal that it need not go on the User Chain. Instead, the Transaction will switch the light to red, then move ahead in the model without delay.

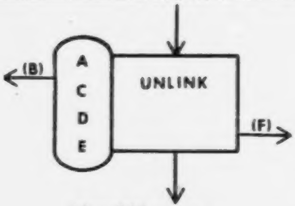
Control of a User Chain's Link Indicator can be summarized this way.

(1) The Link Indicator can be turned "on" (but never turned "off") at the LINK Block.

(2) The Link Indicator can be turned "off" (but never turned "on") at the UNLINK Block.

Consider next a chain-oriented interpretation of the way the UNLINK Block works. When a Transaction enters the UNLINK Block, the Processor removes Transactions from the referenced User Chain, one-by-one, placing each Transaction in turn on the Current Events Chain as the last member in its Priority Class. The Processor works from the front of the User Chain toward the back, unless the D-E Operand combination is "BACK; not used", in which case it works from the back toward the front. Execution of the UNLINK Block causes the Status Change Flag to be turned "on" if at least one Transaction is thereby unlinked [e]. When the UNLINK operation is complete, the Unlinker continues its forward movement in the model. This means that the Unlinked Transactions, if any, have not yet been processed. When the Unlinker finally comes to rest, the Processor tests the Status Change Flag and, if it is "on", turns it "off", and re-starts the scan of the Current Events Chain. This guarantees that, independent of their Priority Level, any Unlinked Transactions will be processed at the current reading of the simulation clock.

Finally, the relationship between User Chains and Block Counts should be carefully noted. When a Transaction is on a User Chain, it is not "in" any Block in the model. In particular, it is not "in" the LINK Block via which it was put onto the User Chain. Transactions on User Chains do not reflect themselves, then, in any fashion through Current Counts at Blocks. When a Transaction has just been unlinked from a User Chain and brought to the Current Events Chain, it would seem that it is also not yet "in" any Block. Conceptually,

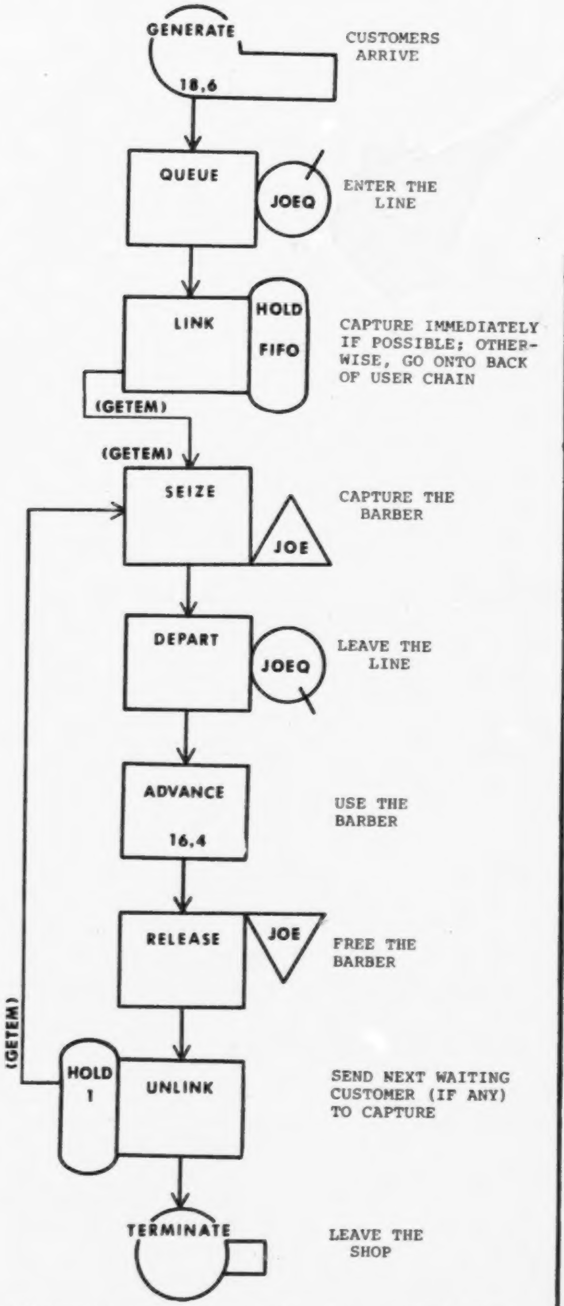


<table>
<tr><th>Operand</th><th>Significance</th><th>Default Value or Result</th></tr>
<tr><td>A</td><td>Name (numeric or symbolic) of a User Chain</td><td>Error</td></tr>
<tr><td>B</td><td>Block Location to which the unlinked Transaction(s) is (are) to be routed</td><td>Error</td></tr>
<tr><td>C</td><td>The number of Transactions to be unlinked (the Unlink Count); can be a constant, a Standard Numerical Attribute, or ALL</td><td>Error</td></tr>
<tr><td>D and E</td><td>Specify which end of the User Chain Transactions are to be taken from, per this scheme:
<table>
<tr><th>D Operand</th><th>E Operand</th><th>End Indicated</th></tr>
<tr><td>Not Used</td><td>Not Used</td><td>Front End</td></tr>
<tr><td>BACK</td><td>Not Used</td><td>Back End</td></tr>
</table></td><td>Transactions are unlinked from the front of the User Chain</td></tr>
<tr><td>F</td><td>Optional Operand; Block Location to which the Unlinker-Transaction moves next if no Transactions are unlinked</td><td>Unlinker-Transaction unconditionally moves to the sequential Block in the model</td></tr>
</table>

Figure 3 The UNLINK Block and Its Operands

---

[e] Familiarity with the concept of the Status Change Flag is assumed. For an explanation, see sections 7.2 and 7.3 in reference [1].


GENERATE
18,6
CUSTOMERS ARRIVE
QUEUE
JOEQ
ENTER THE LINE
LINK
HOLD
FIFO
CAPTURE IMMEDIATELY IF POSSIBLE; OTHERWISE, GO ONTO BACK OF USER CHAIN
(GETEM)
(GETEM)
SEIZE
JOE
CAPTURE THE BARBER
DEPART
JOEQ
LEAVE THE LINE
ADVANCE
16,4
USE THE BARBER
RELEASE
JOE
FREE THE BARBER
(GETEM)
HOLD
1
UNLINK
SEND NEXT WAITING CUSTOMER (IF ANY) TO CAPTURE
TERMINATE
LEAVE THE SHOP

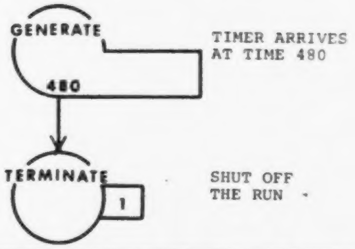

GENERATE
480
TIMER ARRIVES AT TIME 480
TERMINATE
1
SHUT OFF THE RUN

Figure 4 A First Example of User Chain Use

the just-unlinked Transaction has much in common with Transactions which are "on their way" into a model via a GENERATE Block into which they have not yet moved. Nevertheless, from the point of view of Current Counts, the Processor treats unlinked Transactions as though they are in the UNLINK Block whose execution caused them to be brought from the User Chain to the Current Events Chain. This fact is sometimes of importance when Current Block Counts are being interpreted. It also explains why the UNLINK Block's B Operand (i.e., the "Next Block Attempted" for unlinked Transactions) appears in Figure 3 on a path leading from the UNLINK Block. The idea here is to provide a graphic indication of the fact that unlinked Transactions do move from the User Chain via the UNLINK Block into their "Next Block Attempted". In contrast, the other two paths leading from the Figure 3 UNLINK Block apply to the Unlinker-Transaction. One path leads to the sequential Block; the other leads to the non-sequential Block which is implied if the optional F Operand is used.

4. Basic User Chain Use with Facilities and Storages

The basic use of User Chains with Facilities and Storages is illustrated through a series of three examples in this section. First, their use with single Facilities is shown. In this situation, the User Chain Link Indicator is adequate for the required look-ahead logic. Then, their use with Storages is illustrated. Such use requires analyst-supplied look-ahead logic, and this in turn requires caution in terms of a potential simultaneity-of-events problem which can arise. Later, in Sections 5 and 7, additional examples of User Chain use will also be given.

4.1 User Chain Use with a Facility. A Block Diagram for a one-line, one-server queuing system is presented in Figure 4. The particular model shown is for a "one-man barber shop". Interarrival time for customers at the shop is 18±6 minutes; service time is 16±4 minutes. The implicit time unit in the model can consequently be inferred from Figure 4 to be 1 minute. The two-Block timer segment indicates that, when the model is run, the simulation simply shuts off after 480 minutes (i.e., 8 hours) of simulated time. No special provisions are made, then, to provide a realistic closeup feature for the model. Any "customers" in the model at the end of the 8th simulated hour are simply left "as they are". The queue discipline to be practiced in the shop is first-come, first-served.

Notice that a LINK-UNLINK Block pair has been incorporated into the Block Diagram, implying that customer-Transactions who are waiting their turn to get a haircut are kept in this simple model on a User Chain. The LINK Block has been sandwiched between the QUEUE and SEIZE Blocks; similarly, the UNLINK Block is sandwiched between the RELEASE and TERMINATE Blocks. The effect of the presence of these two Blocks will now be explored.

When a customer arrives at the shop, he first updates waiting line statistics by moving into the QUEUE Block. He then moves into the conditional-mode LINK Block. If the Link Indicator is "on" (traffic light red), the customer-Transaction is linked on the back (FIFO) of the User Chain HOLD, and the Link Indicator remains "on". If the Link Indicator is found to be "off" (traffic light green), however, it is switched "on" and the customer-Transaction proceeds to the Block in the location GETEM, i.e., moves into the SEIZE Block. The DEPART-ADVANCE-RELEASE sequence then follows. After the RELEASE, the customer-Transaction attempts to unlink 1 Transaction from the front of the User Chain HOLD (UNLINK Block D and E Operands both blank), sending it to the Block GETEM to capture the now-available Facility. If the attempted unlinking is unsuccessful because the User Chain is empty, the Link Indicator is switched from "on" to "off" (traffic light green) so that the next arrival, instead of linking, will move directly to SEIZE.

Note that, when the traffic light is red at the LINK Block, arriving Transactions are placed on the back of the User Chain (go to the back of the line). Later, via action initiated by an Unlinker Transaction at the UNLINK Block, they are removed from the front of the User Chain. The resulting queue discipline is first-come, first-served [f].

The pattern followed by the Link Indicator in the Figure 4 model reveals how it serves as a built-in look-ahead device in the context of Facility use. It is initially "off". The first customer-Transaction turns it "on", then captures the server. While the server is being used by this first customer of the day, the Link Indicator remains "on". Suppose the second customer-Transaction arrives while the server is still in use. Finding the Link Indicator "on", the second customer goes onto the User Chain. When the first customer finishes, he unlinks the second customer and sends him to capture the barber. Meantime, because the User Chain referenced from the UNLINK Block was not empty, the Link Indicator remains "on". In fact, it is "on" whenever any customer is using the barber, whether that customer (a) found the indicator "off", and moved directly to capture, or (b) found the indicator "on", and spent time in residence on the User Chain before eventually being sent to capture. The only way to turn the Link Indicator "off" is for a customer to finish with the barber when no other customers are waiting (User Chain empty). Turning the Link Indicator "off" in this circumstance guarantees that when the next customer does arrive, he will proceed to capture the barber immeidately.

The punchcards for the Figure 4 model were prepared, and the model was run for one simulated day. The D Operand on the START Card was used to force a chain printout at the end of the simulation. Figure 5 shows a portion of the output that was thereby produced. Parts (a), (b), and (c) in Figure 5 show the Current, Future, and User Chains, respectively. There is a single resident on the Current Events Chain, Transaction 3; this Transaction is poised to release the Facility. [The NBA ("Next Block Attempted") column in Figure 5(a) shows a value of 7. This is the Location occupied in the model by the RELEASE Block, as "counting it out" in Figure 4 will show.] The two residents on the Future Events Chain are the incipient Transaction arrivals at the two GENERATE Blocks in the Model. (Their NBA entries are 1 and 10, respectively, which are the Locations occupied by the GENERATE Blocks in the model.)

In Figure 5(c), the User Chain is described as "USER CHAIN 1". The symbolic name HOLD has been made equivalent to the number 1 by the Processor, and this numeric equivalent has been used to label the User Chain in the printout. There is one Transaction resident on the User Chain, Transaction 4. Note that the various column labels for the User Chain are identical to those for the Current and Future Events Chains.

The Transaction on the User Chain is the next customer, waiting for the barber. We know this because of the problem context, but the GPSS Processor does not know this. In fact, the "destination" of the Transaction on the User Chain will not be known to the Processor until it is unlinked. At that time, the UNLINK Block's B Ope-

---

[f] It is sometimes mistakenly concluded that if the B Operand at a LINK Block is FIFO (meaning that incoming Transactions are linked onto the back of the User Chain), it must be specified at the associated UNLINK Block that Transactions are to be removed from the front of the User Chain; or, that if the B Operand at a LINK Block is LIFO (meaning that incoming Transactions are linked onto the front of the User Chain), the associated UNLINK Block must specify that Transactions are to be removed from the back of the pertinent User Chain. This is not the case. The LINK and UNLINK Blocks are entirely independent of each other. It is the analyst's responsibility to see to it that the linking and unlinking criteria interact in such a way that the overall effect "makes sense" in context. For example, the B Operand at a LINK Block can be FIFO, and the associated UNLINK Block can specify that Transactions are to be unlinked from the back of the pertinent User Chain. The resulting queue discipline would be "last-come, first-served".

```
CURRENT EVENTS CHAIN
TRANS          BDT   BLOCK    PR   SF     NBA     SET    MARK-TIME          P1            P2            P3            P4
    3          480       6                  7       3          453           0             0             0             0
                                                                             0             0             0             0
                                                                             0             0             0             0
```

(a) Current Events Chain*

```
FUTURE EVENTS CHAIN
TRANS          BDT   BLOCK    PR   SF     NBA     SET    MARK-TIME          P1            P2            P3            P4
    1          489                          1       1          -27           0             0             0             0
                                                                             0             0             0             0
                                                                             0             0             0             0
    5          960                         10       5           -1           0             0             0             0
                                                                             0             0             0             0
                                                                             0             0             0             0
```

(b) Future Events Chain*

```
USER CHAIN    1
TRANS          BDT   BLOCK    PR   SF     NBA     SET    MARK-TIME          P1            P2            P3            P4
    4          472                                  4          472           0             0             0             0
                                                                             0             0             0             0
                                                                             0             0             0             0
```

(c) User Chain*

```
USER CHAIN          TOTAL          AVERAGE          CURRENT          AVERAGE          MAXIMUM
                  ENTRIES       TIME/TRANS         CONTENTS         CONTENTS         CONTENTS
    HOLD               18            4.277                1             .160                1
```

(d) User Chain Statistics

```
QUEUE        MAXIMUM     AVERAGE      TOTAL       ZERO     PERCENT       AVERAGE      $AVERAGE    TABLE     CURRENT
            CONTENTS    CONTENTS    ENTRIES    ENTRIES       ZEROS    TIME/TRANS    TIME/TRANS   NUMBER    CONTENTS
   JOEQ            1        .160         27         12        44.4         2.851         5.133                    1
$AVERAGE TIME/TRANS = AVERAGE TIME/TRANS EXCLUDING ZERO ENTRIES
```

(e) Queue Statistics

Figure 5 Selected Output Produced by the Figure 4 Model at the End of the Simulation

*The 7 rightmost columns of information associated with these chains have been eliminated.

rand will be used by the Processor to determine the unlinked Transaction's "Next Block Attempted". Note, then, the entry in the BLOCK column in Figure 5(c) is "blank". The BLOCK column indicates which Block a Transaction is currently "in". But, as explained earlier, when a Transaction is on a User Chain, it is not "in" any Block in the model.

The BDT ("Block Departure Time") column in Figure 5(c) shows a value of 472. Block Departure Time is the time the Transaction is scheduled to try to move into its "Next Block Attempted." As far as its "future movement" is concerned, the BDT entry for User Chain Transactions is meaningless. The BDT value shown in User Chain printout can be interpreted as the time the Transaction was linked onto the User Chain.

The statistics for the User Chain HOLD which appear in the standard output are shown in Figure 5(d). Figure 5(e) shows the statistics for the Queue JOEQ. Comparison of the two sets of staticstics reveals that they are quite similar. The Queue statistics contain somewhat more information than those for the User Chain, indicating how many zero entries there were (ZERO ENTRIES), what percentage the zero entries were of the total (PERCENT ZEROS), and what the average Queue residence time was when zero entries were included (AVERAGE TIME/TRANS).

At first, it might be thought that there are no "zero entries" to User Chains because, "if blockage does not exist, Transactions bypass the chain and move directly forward in the model." User Chains can experience zero entries, however. That is, it is possible for some Transactions to have zero residence time on a User Chain. This will happen, for example, in the Figure 4 model when the following conditions are true.

(1) The Facility is in use.

(2) No Transaction is waiting to capture the Facility.

(3) There is a time-tie between the two events "completion of service", and "arrival of the next customer".

(4) The event-sequence is "arrival", followed by "service completion".

In the scan of the Current Events Chain at the simulated time in question, then, the arriving customer-Transaction is processed first, per (4) above. Finding the User Chain's Link Indicator "on", the Processor puts this Transaction on the User Chain. The releasing Transaction is then processed. After moving through the RELEASE Block, it unlinks the just-arrived Transaction from the User Chain and sends it to capture the Facility. Hence, although the just-arrived Transaction was made a User Chain resident, its residence time on the chain was zero. It contributes then, to the User Chain TOTAL ENTRIES statistic. And, from the Queue's point of view, it contributes to the ZERO ENTRIES statistic.

The phenomenon just described explains why there were 18 TOTAL ENTRIES to the User Chain in Figure 5(d), but only 15 non-zero entries to the Queue in Figure 5(e). Three of the User Chain entries were apparently of the "zero residence time" type. Note that this phenomenon also makes interpretation of the AVERAGE TIME/TRANS statistic for User Chains somewhat subtle. It would be easy to draw the false conclusion for the Figure 4 model that $AVERAGE TIME/TRANS in the Queue should equal the AVERAGE TIME/TRANS statistic for the User Chain. $AVERAGE TIME/TRANS measures the waiting time only of those who had to wait, however; in contrast, the AVERAGE TIME/TRANS value for User Chains can, in general, include Transactions which did not actually have to wait. The 3 "zero residence time" entries to the User Chain explains why AVERAGE TIME/TRANS is only 4.277 time units in Figure 5(d), whereas $AVERAGE TIME/TRANS is 5.1333 time units in Figure 5(e).

User Chain statistics and Queue statistics, although similar, differ from each other, then, in these three major ways.

(1) Zero-entry information is provided for Queues.

(2) The AVERAGE TIME/TRANS User Chain statistic requires careful interpretation.

(3) The distribution of Queue residence time is easily estimated with use of the QTABLE Card, whereas nothing analogous to the QTABLE Card is available for User Chains.

4.2 More About the Link Indicator. Consider use of the LINK-UNLINK Block pair in connection with any segment of a GPSS Block Diagram. Figure 6 illustrates this situation where, for generality, the particular Blocks occupying the Block Diagram segment in question are not shown, but for specificity the LINK-UNLINK Block Operands are shown. Assume that Transactions can gain entry to the segment only by moving through the conditional-mode LINK Block in the figure, and that they can exit the segment only by moving through the UNLINK Block. There can then never be more than one Transaction in the encircled Block Diagram segment at a time.

This last statement can be made as a direct consequence of the properties of the User Chain's Link Indicator. The reasoning goes like this. When the simulation starts, the encircled segment is empty. Furthermore, the Link Indicator is "off". When the first Transaction arrives at the LINK Block, it therefore moves immediately to the Block in the Location MOVIN, thereby entering the segment. (MOVIN is assumed to be the Location of the "first" Block in the encircled segment.) While the first Transaction is in the segment, then, any other arrivals at the LINK Block are put on the User Chain. When the first Transaction eventually leaves the segment via the UNLINK Block, it unhooks exactly 1 Transaction from the User Chain, routing it into the segment.

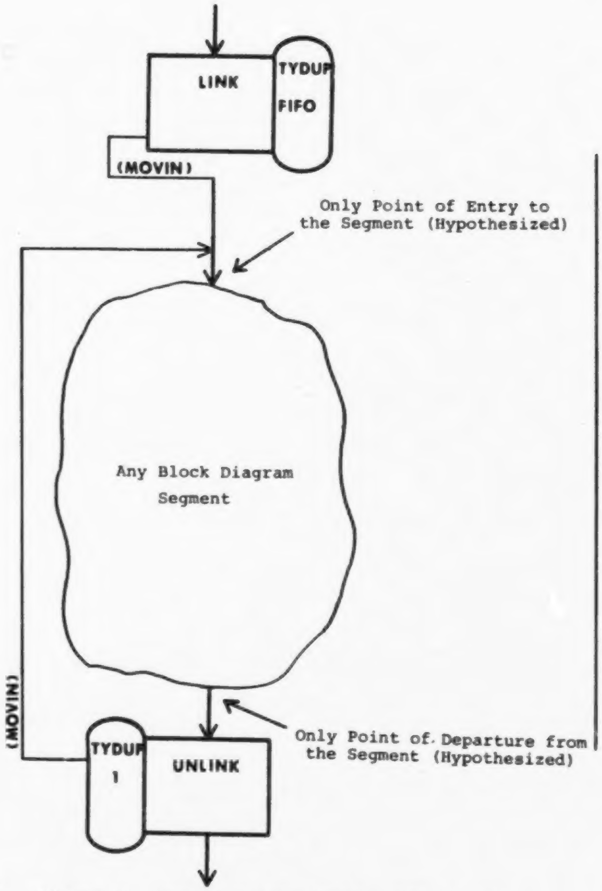

LINK
TYDUP
FIFO
(MOVIN)
Only Point of Entry to the Segment (Hypothesized)
Any Block Diagram Segment
(MOVIN)
TYDUP
1
UNLINK
Only Point of Departure from the Segment (Hypothesized)

Figure 6 Use of a User Chain with an Arbitrary Block Diagram Segment

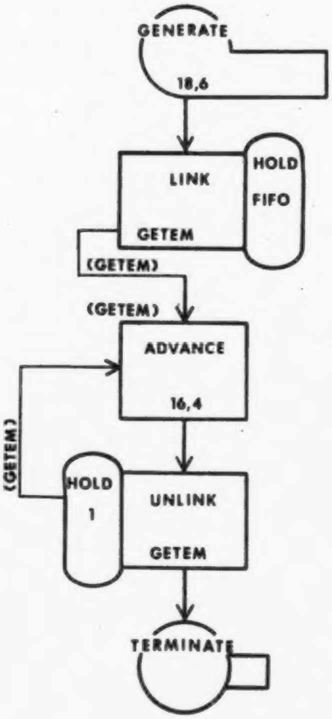

GENERATE
18,6
LINK
HOLD
FIFO
GETEM
(GETEM)
(GETEM)
ADVANCE
16,4
(GETEM)
HOLD
1
UNLINK
GETEM
TERMINATE

Figure 7 A Model Segment which Simulates a Single Server without Using a SEIZE-RELEASE Block Pair

Hence, the segment-exiting Transaction "replaces itself" in the segment with another Transaction. This "replacement pattern" is in effect as long as there is at least 1 Transaction on the User Chain when the UNLINK Block is executed. If the User Chain is empty when the UNLINK Block is executed, the result is that the Link Indicator gets turned "off". This means that when another Transaction eventually arrives at the LINK Block, it moves immediately into the segment, causing the Link Indicator to be turned back "on" in the process, etc.,etc.

The ideas just expressed really only repeat what was said about the Link Indicator when it was introduced in Section 3. Repeating the ideas in the context of Figure 6, however, leads directly to the two following conclusions.

Conclusion 1. When a User Chain is used in connection with a Facility, the SEIZE-RELEASE Block pair is not really needed, unless the analyst requires the statistics which the Facility entity provides. After all, use of a SEIZE-RELEASE Block pair has two effects.

(1) It guarantees that there will never be more than one Transaction at a time in transit between the pair of Blocks (assuming, of course, that alternative methods of "getting between" the pair of Blocks are not used in the model).

(2) It causes the GPSS Processor to maintain certain statistics about the "use" of the Facility.

But Effect (1) is precisely the effect that the Link Indicator has when a conditional-mode LINK Block is used. Consequently, if Effect (2) is not needed, the SEIZE-RELEASE Block pair can be eliminated. For example, Figure 7 repeats Figure 4, with the SEIZE-RELEASE Block pair eliminated. The QUEUE-DEPART Block pair has also been eliminated, on the hypothesis that the User Chain statistics are sufficient measures of waiting line behavior for the application at hand. The Block sequence "LINK-ADVANCE-UNLINK" in Figure 7 may seem a bit strange at first, but it nonetheless validly simulates a single server under the conditions stated here.

Conclusion 2. When the User Chain's Link Indicator is relied upon to supply "look-ahead logic", it is extremely inflexible. In fact, because a consequence of its use is to let only "one Transaction at a time" in the model segment between the LINK and UNLINK Blocks, the Link Indicator is really only of value when the constrained resource being simulated between the LINK-UNLINK Blocks is a Facility (or a unit-capacity constraint). For example, suppose the constrained resource is being simulated with a Storage whose capacity is two. This means that up to 2 Transactions at a time should be permitted to be in transit between the LINK-UNLINK Block pair. Because this effect cannot be achieved with the Link Indicator, the analyst must supply his own look-ahead logic to determine whether an arriving Transaction can move into the model segment, or must be put onto the User Chain. The next section goes into further detail about use of a User Chain with the Storage entity.

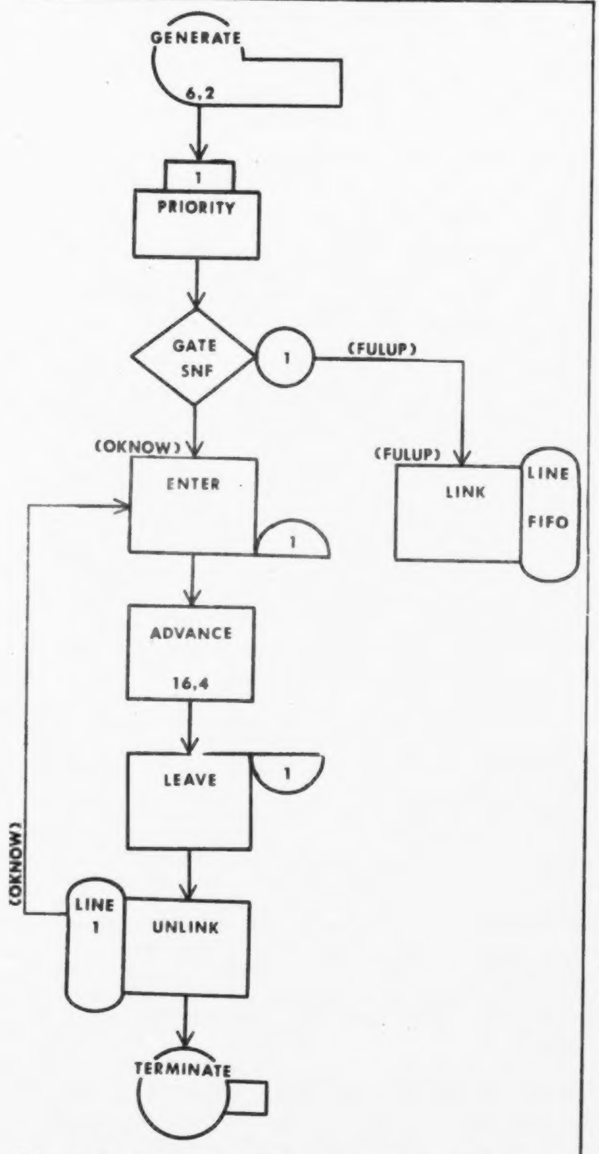


Figure 8 A Second Example of User Chain Use

4.3 User Chain User with a Storage. Suppose that in the Figure 4 barber shop, customer inter-arrival time decreases to 6+2 minutes and, to offset this heavier traffic pattern, two more barbers are hired. Figure 8 shows the Block Diagram for a model of the shop under these circumstances. Discussion of the model will be broken into two parts. First, the "GATE-ENTER-LINK" Block arrangement will be commented upon. Then the reason for placing the PRIORITY Block between the GENERATE and GATE Blocks will be explained.

As indicated under Conclusion 2 in the preceding sub-section, the analyst must supply his own look-ahead logic when a User Chain is used in conjunction with a Storage. The GATE Block in Figure 8 provides this required look-ahead logic. When a customer-Transaction moves into the "GATE SNF 1" Block, a test is conducted to determine whether at least one barber is currently available, i.e., to determine whether the Storage used to simulate the three barbers is not full. If the "Storage Not Full" condition is true, the customer-Transaction moves sequentially through the gate and captures a barber. If the "Storage Not Full" condition is false, the customer-Transaction exits the gate non-sequentially and moves into the LINK Block. No C Operand is provided with the LINK Block, with the result that Transactions entering it are unconditionally placed on the User Chain. Transactions enter the LINK Block, however, only on the condition that the Storage is full. Via use of the GATE Block, then, the "unconditional" linking of Transactions is forced to be conditional after all.

Now consider why the "PRIORITY 1" Block has been placed between the GENERATE and GATE Blocks in the Figure 8 model. The PRIORITY Block has been used to defense against invalid logic which could come about if a certain simultaneity-of-events situation were to arise. Suppose that the following conditions are true at a given point in simulated time.

(1) All 3 barbers are captured.
(2) At least 1 Transaction is waiting on the User Chain.
(3) One of the in-service customers is just leaving.
(4) The next customer is just arriving.
(5) The leaving Transaction is ahead of the arriving Transaction on the Current Events Chain.

When the leaving Transaction is processed, it first moves into the LEAVE Block, thereby changing the condition "Storage Not Full" from false to true. This leaving Transaction then moves into the UNLINK Block, causing the Transaction at the front of the User Chain to be moved to the Current Events Chain. Because of its earlier movement through the PRIORITY Block, the unlinked

Transaction has a Priority Level of 1. It is therefore placed on the Current Events Chain ahead of the arriving customer-Transaction, which has a Priority Level of 0. After the leaving Transaction has terminated, the Processor restarts the CEC scan. It first processes the just-unlinked customer-Transaction, moving it into the ENTER-ADVANCE sequence. Execution of the ENTER Block results in the condition "Storage Not Full" being made false again. When the arriving customer-Transaction is processed later in the scan, it therefore moves non-sequentially from the GATE Block to the LINK Block, and is put on the User Chain. The previously-waiting customer has captured the barber, which is as it should be.

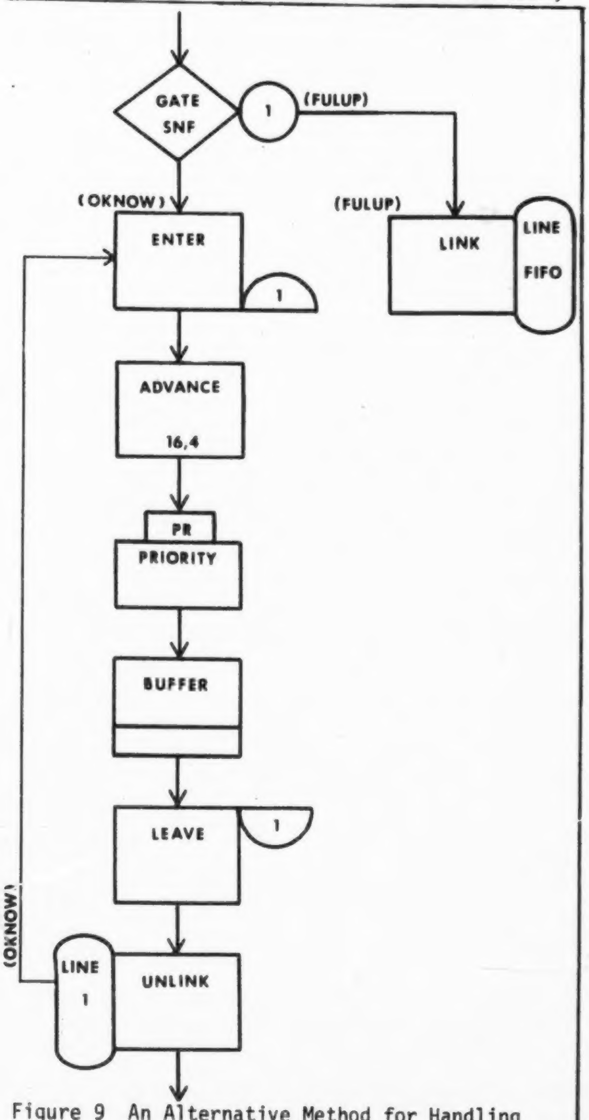


Figure 9 An Alternative Method for Handling the Potential Problem of Simultaneity in the Figure 8 Model

It is easy to see how the logic of the model would be subverted if the PRIORITY Block were removed from the model, and the conditions described above came about. The unlinked Transaction would be put on the current Events Chain behind the just-arriving customer-Transaction. When the arriving Transaction reached the GATE Block, the "Storage Not Full" condition would be true. The "newcomer" would therefore capture the barber. When the just-unlinked Transaction eventually tried to move into the ENTER Block, entry would be denied. This previously-waiting Transaction would have "missed its chance". Furthermore, its subsequent waiting would take place on the Current Events Chain, not on the User Chain.

Whether using the PRIORITY Block in the Figure 8 model is "worth it" can be debated. The Block does guarantee that the logic of the model will always be valid. On the other hand, to include this "additional" Block increases the number of Blocks in the model from 8 to 9. Speaking very roughly, this means that execution time for the model is increased by about 12.5% relative to the no-PRIORITY-Block version of the model [g]. Depending on the size of the implicit time unit and the intensity of the traffic pattern, the conditions leading to the "problem of simultaneity" may come about very infrequently. To save execution time in modeling situations such as this, the analyst might prefer to deliberately exclude the PRIORITY Block, and simply "accept" occasional invalidity in his models. (Note that the potential "problem of simultaneity" did not have to be defensed against when the Link Indicator provided the look-ahead logic in Figure 4. To check your understanding of the ideas presented so far in this paper, you should now try to explain why this is true).

4.4 An Alternative for Handling the Simultaneity Problem. In the Figure 8 model, it was "convenient" to place the PRIORITY Block between the GENERATE and GATE Blocks. But this was largely because the modeling context used as an example was completely self-contained. It is not normally true that a Transaction "approaches" a Storage from a GENERATE Block. More often than not, the "approach" is made from some preceding non-trivial model segment. The question then arises, how is one to handle the problem of simultaneity in this circumstance? The purpose of this section is to suggest an alternative which can be used under more complicated circumstances.

Figure 9 shows the suggested alternative in the

---

[g] When the PRIORITY Block's A Operand is a constant, a minimum of 114 assembler instruction-executions are performed when a Transaction moves into the Block. (This count applies to GPSS/360, Version 1, Modification Level 3.)

form of a Block Diagram segment corresponding to use of the Figure 8 Storage. It is assumed that all Transactions move into the segment with a common Priority Level, whatever that may be. The "defense" against the simultaneity problem takes the form of the PRIORITY-BUFFER sequence placed between the ADVANCE and LEAVE Blocks.

Consider how the PRIORITY-BUFFER combination works to eliminate the simultaneity problem. Assume that the conditions required for the simultaneity problem are in effect, as follows.

(1) All 3 barbers are captured.

(2) At least 1 Transaction is waiting on the User Chain.

(3) One of the in-service customers is just leaving.

(4) The next customer is just arriving.

(5) The leaving Transaction is ahead of the arriving Transaction on the Current Events Chain.

Now, when the leaving Transaction is processed, it immediately moves into the PRIORITY Block, where its "old" Priority Level is reassigned as its "new" Priority Level. This produces no change in Priority Level, but it does cause the Processor to re-position the Transaction on the Current Events Chain as the last member in its "new" Priority Class. This means that the leaving Transaction is now behind the arriving Transaction (which reverses condition (5) stated above). The leaving Transaction then moves into the BUFFER Block, forcing the Processor to re-start its CEC scan. As the re-initiated scan proceeds, the arriving Transaction is (eventually) encountered, finds the condition "Storage Not Full" is false (the leaving Transaction has not yet executed the LEAVE Block), and therefore transfers non-sequentially to the User Chain. Later in the scan, the Processor resumes the forward movement of the leaving Transaction, moving it through the LEAVE Block to the UNLINK Block. The Transaction at the front of the User Chain is then transferred to the Current Events Chain, and enters the Storage when the scan is re-started (execution of the LEAVE Block caused the Status Change Flag to be turned "on"; execution of the UNLINK Block then caused it to be turned on "again", redundantly).

It should be clear what would happen under the stated conditions if the PRIORITY-BUFFER Blocks were not in the model. The leaving Transaction would be processed first, making the condition "Storage Not Full" true. The unlinked Transaction would be put on the Current Events Chain behind the arriving Transaction, since they are postulated to have the same Priority Level. The arriving Transaction would then enter the Storage, thereby capturing the server who had been intended for the unlinked Transaction. By the time the unlinked Transaction was processed, there would be "no room left" for it in the Storage. In short, the logic of the model would be invalid.

In Figure 9, the PRIORITY and BUFFER Blocks were shown separately, albeit in sequence, to make the preceding explanation a bit more straightforward. Active GPSS users will recall that when the "buffer option" is used with the PRIORITY Block, the effect achieved is precisely identical to that of the two-Block PRIORITY-BUFFER sequence shown in Figure 9. That is, the single Block "PRIORITY PR,BUFFER" could be used to replace the PRIORITY-BUFFER Block-pair. In the remaining examples in this paper, this "buffer option" will be used with the PRIORITY Block when the occasion arises, for the sake of "Block economy".

5. More Examples of User Chain Use

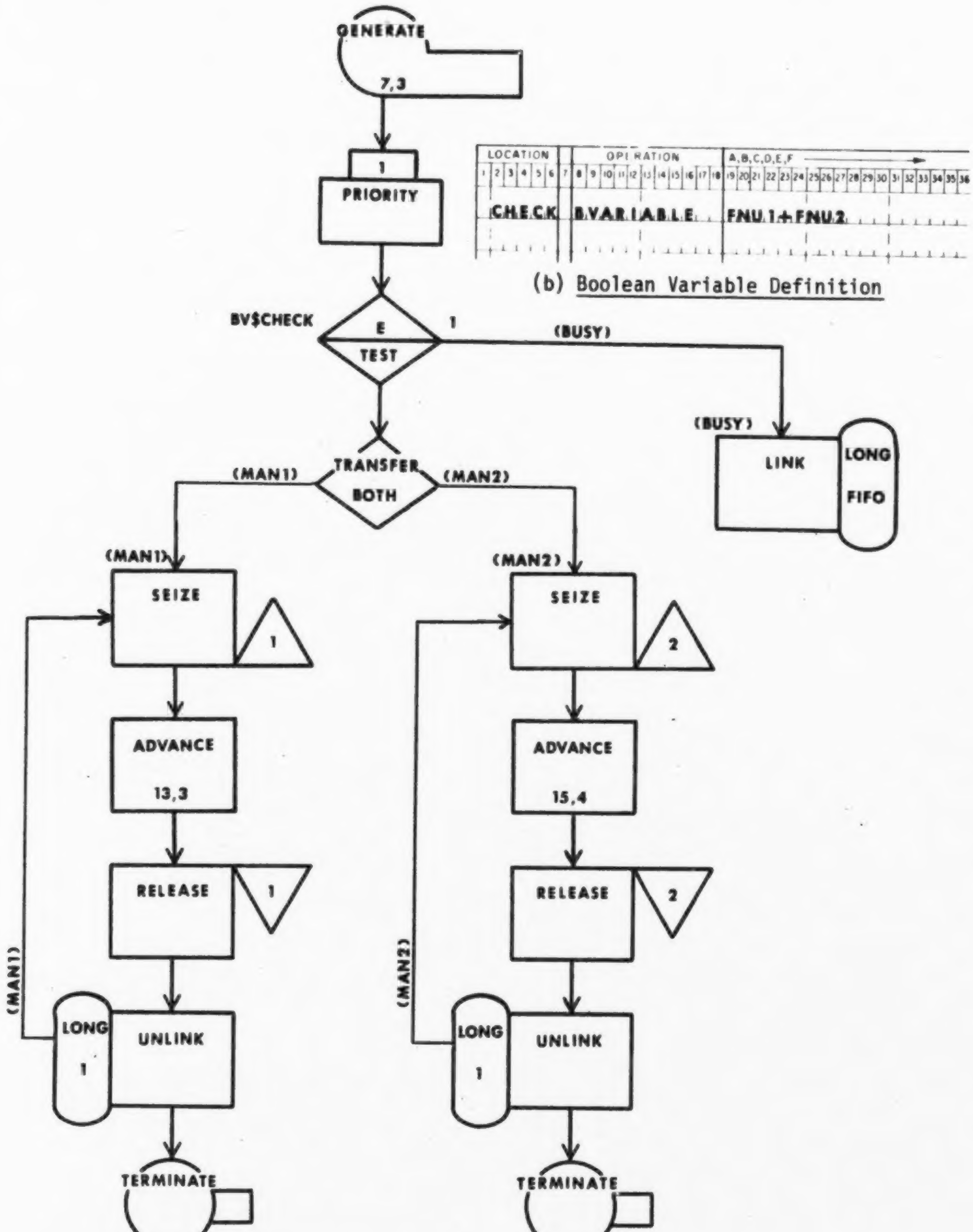
Two more examples of User Chain use with Facilities are given in this section. The first example illustrates application of User Chains with two Facilities operating in parallel. The second example shows how the "shortest imminent operation" queue discipline is simulated with use of Parameter-mode linking at the LINK Block.

5.1 User Chain Use with Two Parallel Facilities.

Suppose that two barbers work in a barber shop. Service time for the first and second of these barbers is 13±3 and 15±4 minutes, respectively. Customers arrive at the shop every 7±3 minutes. Figure 10 shows how a User Chain can be incorporated into a model of the barber shop. When a customer-Transaction arrives at the shop, it moves (through the PRIORITY Block) into the TEST Block shown in Figure 10(a). There, it evaluates the Boolean Variable CHECK, as defined in Figure 10(b), to determine whether either one or both of the barbers are available. If a barber is free, the customer-Transaction proceeds to the TRANSFER Block in BOTH mode, from whence it is routed to a SEIZE Block not currently denying entry. If both barbers are busy, it is routed from the transfer-mode TEST Block to the unconditional-mode LINK Block. There, it is linked onto the back of the User Chain LONG.

Whenever a customer-Transaction releases a barber, it causes another waiting customer to be unlinked from the User Chain and routed directly to the pertinent SEIZE Block. Because of the "PRIORITY 1" Block following the GENERATE Block, the unlinked customer-Transaction is assured of being the one to capture the just-released barber under all circumstances.

This example is one in which the execution time savings resulting from User Chain use can be substantial. In contrast with Figures 4 and 8, where the potential blocking conditions are both unique, the TRANSFER Block in Figure 10 offers non-unique blocking. If the User Chain were not used, each delayed Transaction on the Current Events Chain would attempt to move through the TRANSFER Block at each CEC scan, thereby consuming a telling amount of time. For the model shown, the TRANSFER Block is executed successfully one time by each arriving customer who finds a barber immediately available. No attempt is otherwise made to execute the Block.



LOCATION	OPERATION	A,B,C,D,E,F
CHECK	BVARIABLE	FNU1+FNU2

(b) Boolean Variable Definition

GENERATE
7,3
1
PRIORITY
BV$CHECK
E
TEST
1
(BUSY)
(BUSY)
LINK
LONG
FIFO
TRANSFER
BOTH
(MAN1)
(MAN2)
(MAN1)
(MAN2)
SEIZE
1
SEIZE
2
ADVANCE
13,3
ADVANCE
15,4
RELEASE
1
RELEASE
2
(MAN1)
(MAN2)
LONG
1
UNLINK
LONG
1
UNLINK
TERMINATE
TERMINATE

(a) Block Diagram

Figure 10 A Third Example of User Chain Use

5.2 User Chain Use for "Shortest Imminent Operation" Queue Discipline. Consider this problem. At the Facility MAC, the queue discipline practiced is "shortest imminent operation". This means that the Transaction expected to hold the Facility the shortest length of time is the one permitted to capture it next. In case of ties for shortest imminent operation, the ties are to be resolved on a first-come, first-served basis. When a Transaction does capture the Facility, its actual holding time follows the exponential distribution.

Assuming that a Transaction's expected holding time at the Facility is stored in its second Parameter, Figure 11 shows a Block Diagram segment which implements this queue discipline. (It is assumed that the Function symbolically named XPDIS is the usual 24-point Function used to sample from an exponential distribution with a mean of 1.) Transactions waiting for the Facility are put onto the User Chain QUE, ordered according to their P2 value. This means they are put onto the chain in order of "shortest imminent operation", with expected operation times increasing from the front of the User Chain toward the back. Furthermore, in event of ties, each most-recent arrival is placed behind earlier arrivals which have the same P2 value. In short, linking "in Parameter mode" results in direct implementation of the shortest imminent operation queue discipline (assuming, of course, that Transactions are later unhooked from the front of the User Chain).

In the Figure 11 model segment, just before a Transaction releases the Facility, it moves into a "PRIORITY PR,BUFFER" Block. The Processor therefore re-positions the Transaction on the Current Events Chain as the last member in its Priority Class, and re-starts the scan. This guarantees that, in case of a time-tie between the events "next arrival of a job", and "release of the Facility", the arriving job-Transaction is hooked onto the User Chain before the next Transaction to capture the Facility is captured. (It is assumed that all Transactions which use the Facility have the same Priority Level.) The result is to insure that the arriving job-Transaction is included in the "competition" that takes place to see which waiting job has the shortest imminent operation. If this simultaneity-of-events situation arose and the PRIORITY Block were not included in the Figure 11 model segment, the shortest imminent operation queue discipline could be violated.

6. User Chain Standard Numerical Attributes

Each User Chain has five Standard Numerical Attributes associated with it. The pre-defined names of these attributes, and the significance of the corresponding values, are shown in Table 1.

When the Processor encounters a CLEAR Card, the values of these Standard Numerical Attributes are set to zero, and any User Chain residents are removed from the model. The effect of the RESET Card is to set the values of CAj, CCj, and CTj to zero. The value of CHj remains the same, and the value of CMj is set to the current value of CHj. Of course, any User Chain residents are left undisturbed during the resetting operation.

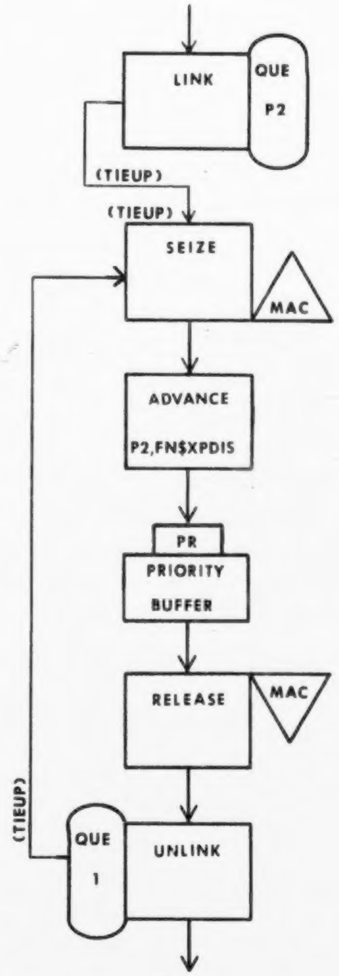


Figure 11 A Fourth Example of User Chain Use

7. Conditional Unlinking of Transactions from User Chains

In Section 3, use of the UNLINK Block D and E Operands to remove Transactions from (a) the front, or (b) the back of User Chains was introduced. In neither of these cases does a Transaction have to meet a particular condition, other than "relative position on the chain", to qualify for unlinking. There are three other D and E Operand combinations which can be used to impose on potential Unlinkee Transactions the require-

Pre-defined Name [h]	Value
CAj, or CA$sn	Integer portion of the average number of Transactions on the chain
CCj, or CC$sn	The total number of Transactions hooked onto the chain during the course of the simulation
CHj, or CH$sn	The number of Transactions currently on the chain
CMj, or CM$sn	Maximum number of Transactions simultaneously resident on the chain during the simulation; the maximum value CHj (or CH$sn) has attained
CTj, or CT$sn	Integer portion of the average Transaction residence time on the chain

Table 1 User Chain Standard Numerical Attributes

Combination Number	D Operand	E Operand	Condition Required for Unlinking
1	Any Standard Numerical Attribute	Not used	Let "j" represent the value of the D Operand; the User Chain Transaction qualifies for unlinking if its j-th Parameter value equals the value of the Unlinker's j-th Parameter
2	Any Standard Numerical Attribute	Any Standard Numerical Attribute	Let "j" represent the value of the D Operand; the potential Unlinkee qualifies if its j-th Parameter value equals the value of the E Operand
3	BVj, or BV$sn	Not used	The potential Unlinkee qualifies if the Boolean Variable numbered j (or symbolically named sn) is true when it is evaluated with that Transaction's Priority Level and Parameter values

Table 2 Additional D and E Combinations Possible for the UNLINK Block

ment that they satisfy a specified condition. These other three combinations are shown in Table 2.

For all three of the Table 2 combinations, the User Chain is scanned from front to back by the Processor until the Unlink Count has been satisfied, or the back of the chain has been reached, whichever occurs first. In Combination 1, the value of a specified Parameter of the potential Unlinkee must equal the value of the same Parameter of the Unlinker. The UNLINK Block's D Operand indicates the number of the applicable Parameter; the E Operand is not used. In Combination 2, the value of a specified Parameter of the potential Unlinkee must equal some other arbitrarily-specified value. The UNLINK Block's D Operand again provides the number of the potential Unlinkee's applicable Parameter; the E Operand is the "Match Argument", i.e., provides the value which the Unlinkee's Parameter value must equal.

In Combination 3, the D Operand references a Boolean Variable, and the E Operand is not used. For each Transaction on the User Chain, the Processor evaluates the Boolean Variable. Only if its value is true does the User Chain Transaction qualify for unlinking. The question naturally arises, "how can the value of a Boolean Variable be made to depend on properties of a Transaction on a User Chain?" The answer is that if numeric data references in the Boolean Variable include Priority Level and/or Parameter values, the User Chain Transaction currently being examined supplies these values, not the Transaction at the UNLINK Block.

An example will now be given to show use of a Boolean Variable with the UNLINK Block. Consider the "shortest imminent operation" queue discipline, as illustrated in Figure 11. A disadvantage of this queue discipline is that jobs with a large imminent operation time can be delayed for very long times waiting for the Facility. This happens if jobs with shorter operation times keep arriving at the Facility before the bigger jobs can capture it. The problem can be avoided by dividing all waiting jobs into two groups, as determined by how long they have been waiting. Highest priority is given to those jobs that have been waiting longer than some pre-determined time, called the critical threshhold. Jobs in this group are termed "critical". Within the set of critical jobs, queue discipline is "shortest imminent operation". Queue discipline for the non-critical jobs is also "shortest imminent operation". The overall queue discipline, then, is "serve critical jobs first, then serve non-critical jobs; in each of these two categories, select jobs according to shortest imminent operation."

---

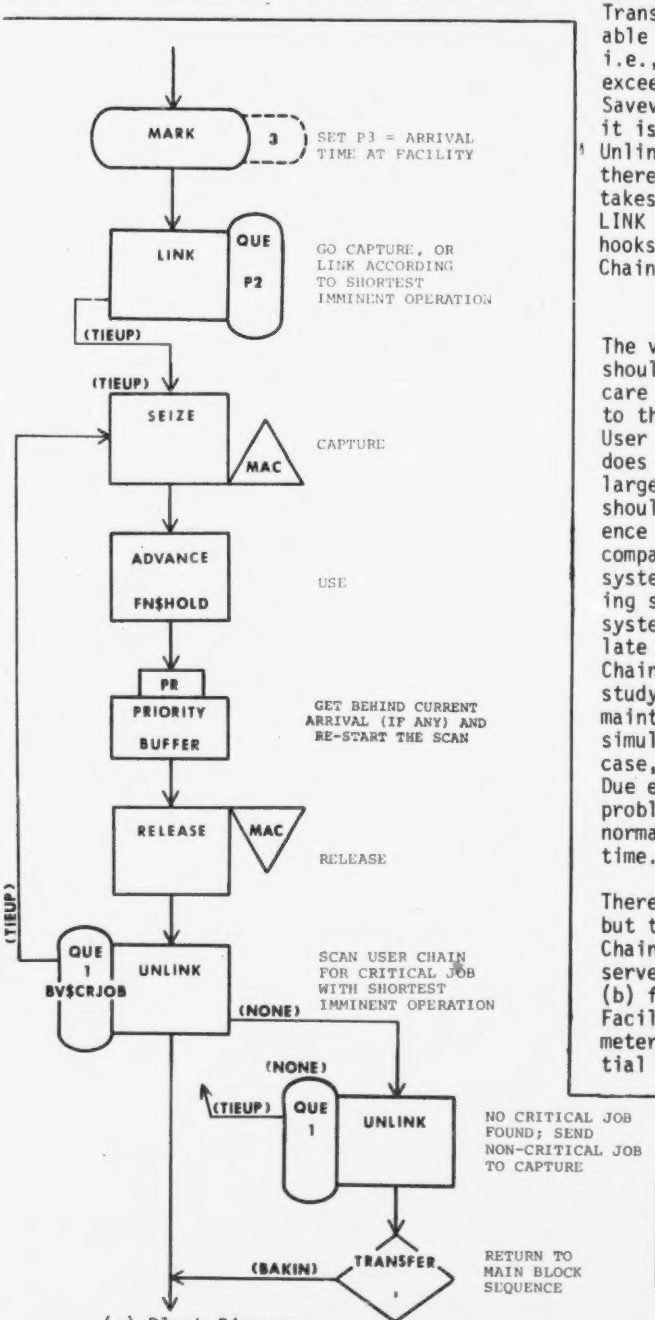
[h] "j" is understood to be the number of the User Chain, if it has been named numerically; "sn" is understood to be its symbolic name, if it has been named symbolically.

A Block Diagram for this overall queue discipline is shown in Figure 12(a). When a job-Transaction enters the segment, its time-of-arrival is first marked in Parameter 3. The Transaction then captures the Facility MAC immediately if possible, and otherwise goes onto the User Chain, ordered according to its imminent operation time as carried in Parameter 2. When a job-Transaction is finished using the Facility, it enters an UNLINK Block to request a Boolean-mode scan of the User Chain. The User Chain is scanned from front-to-back in a search for the first job-Transaction, if any, for which the Boolean Variable CRJOB [defined in Figure 12(b)] is true, i.e., for which residence time on the User Chain exceeds the critical threshhold, as held in the Savevalue CRTYM. If such a Transaction is found, it is unhooked and sent to capture; meantime, the Unlinker continues to the sequential Block. If there are no critical jobs, however, the Unlinker takes the non-sequential exit from the first UNLINK Block to a second UNLINK Block, where it unhooks the front-end Transaction from the User Chain and sends it to capture.

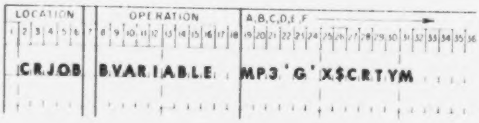
8. Other Sources of Examples

The various examples presented in this paper should alert the GPSS analyst to the fact that care must be exercised, especially with respect to the "simultaneity of events" problem, when User Chains are applied in the language. Space does not permit presentation of additional, larger-scale examples here. Those interested should refer to case studies 7A and 7C in reference [1]. Case study 7A employs User Chains in comparing a one-line, multiple-server queuing system to a multiple-line, multiple-server queuing system in a banking context. For the latter system, Facilities are used in parallel to simulate the parallel servers, and a separate User Chain is associated with each Facility. In case study 7C, in the context of a "city's vehicle-maintenance garage", parallel servers are also simulated with Facilities in parallel. In this case, pre-emptive use of Facilities is allowed. Due emphasis is given to a simultaneity-of-events problem which can arise when pre-emption and a normal "release" of a Facility occur at the same time.

There are examples in other "pedagogical" sources, but they do not abound. In [2], examples of User Chain use are given (a) for first-come, first-served queue discipline with a single Facility, (b) for random queue discipline with a single Facility, (c) when unlinking is based on a "Parameter-match" between the unlinker, and the potential unlinkee, and (d) when a Boolean Variable


MARK
3
SET P3 = ARRIVAL TIME AT FACILITY
LINK
QUE
P2
GO CAPTURE, OR LINK ACCORDING TO SHORTEST IMMINENT OPERATION
(TIEUP)
(TIEUP)
SEIZE
MAC
CAPTURE
ADVANCE
FN$HOLD
USE
PR
PRIORITY
BUFFER
GET BEHIND CURRENT ARRIVAL (IF ANY) AND RE-START THE SCAN
RELEASE
MAC
RELEASE
(TIEUP)
QUE
1
BV$CRJOB
UNLINK
SCAN USER CHAIN FOR CRITICAL JOB WITH SHORTEST IMMINENT OPERATION
(NONE)
(NONE)
(TIEUP)
QUE
1
UNLINK
NO CRITICAL JOB FOUND; SEND NON-CRITICAL JOB TO CAPTURE
TRANSFER
,
(BAKIN)
RETURN TO MAIN BLOCK SEQUENCE

(a) Block Diagram



LOCATION	OPERATION	A,B,C,D,E,F
CRJOB	BVARIABLE	MP3'G'X$CRTYM

(b) Boolean Variable Definition

Figure 12 A Fifth Example of User Chain Use

must be true before unlinking can occur. The same examples are repeated in [3] and [4]. There are no examples in these sources that consider User Chain use for constrained resources simulated with more than a single Facility, and the simultaneity-of-events problem is not mentioned. In [5], a "random queue discipline" model is shown which consumes less CPU time than the one given in the above sources. Use of User Chains to implement "least job slack per remaining operation" queue discipline in a job shop problem is also shown in [5]. In [6], Greenberg gives 6 examples of User Chain use, restricting the constrained resource to a single Facility. The purpose of having 6 examples is to explain variour LINK-UNLINK Block Operand combinations. No additional applications are illustrated, and no mention is made of the simultaneity-of-events problem.

9. Summary

This paper, presented as a tutorial, elaborates on the User Chain entity in GPSS. The concept of User Chains is presented, the potential benefits to be gained from their use are described, and a detailed description of the two GPSS Blocks supporting User Chain use is provided. Examples illustrating a range of User Chain applications are introduced and discussed. The potential "simultaneity-of-events" problem that can occur even with only modestly imaginative User Chain use is brought to light in several of these examples. Reference is made to additional sources of examples in the literature.

10. Biography

Thomas J. Schriber is a Professor of Management Science at the University of Michigan. He regularly teaches a 5-day "introductory" course and a 3-day "advanced" course on GPSS in the University of Michigan's Engineering Summer Conference Series. He gives GPSS courses in industry, and has written two introductory books on the subject.

11. References

[1] Schriber, Thomas J., A GPSS Primer (currently available in softbound form from Ulrich's Books, Inc., Ann Arbor, MI; being published in hardbound by John Wiley & Sons, Inc., with publication date not yet set; title only tentative for hardbound form)

[2] GPSS/360 User's Manual (IBM; Form Number GH20-0326)

[3] GPSS/360 Version 2 User's Manual (IBM; Form Number SH20-0694)

[4] GPSS V User's Manual (IBM; Form Number SH20-0851)

[5] Schriber, Thomas J., GPSS/360: Introductory Concepts and Case Studies (Ulrich's Books, Inc., 1968; 1969; 1971)

[6] Greenberg, Stanley, GPSS Primer (Wiley-Interscience, 1972)

Session 4: Health Services

Chairman: Dean Uyeno, University of British Columbia

Simulation in the health services has seen increasing use. While many papers have been written, only a few have cited the monetary savings which the application of the technique has created. Two of the papers in this session fall into this latter category. The third paper considers the use of health auxiliaries to alleviate pressing health manpower problems.

Papers

"An Evaluation of Expanded Function Auxiliaries in General Dentiestry"
Kerry E. Kilpatrick and Richard S. Mackenzie, University of Florida

"The Use of Computer Simulation in Health Care Planning"
O. George Kennedy, MEDICUS Systems Corporation

"A Simulation Model of a University Health Service Outpatient Clinic"
Robert Baron and Edward Rising, University of Massachusetts

Discussants

Hal Jackman, Secretariat for Social Development (Ontario)
Joseph Shartiag, Chicago Health Services Research Center

AN EVALUATION OF EXPANDED FUNCTION AUXILIARIES IN GENERAL DENTISTRY

Kerry E. Kilpatrick, Richard S. Mackenzie, and Allen G. Delaney
Health Systems Research Division, University of Florida, Gainesville, Florida

Abstract

A simulation model of private dental practice has been developed to evaluate the effects of introducing expanded function auxiliary personnel. The model permits the experimental investigation of a variety of staffing patterns and facility configurations. Results of these experiments indicate that a solo practice can expand its patient volume 169% and increase net revenue by 233% by adding expanded function auxiliaries while simultaneously reducing patient waiting time and the time spent at chairside by the dentist. Field validations of the simulation results are described.

All segments of the American health care system are confronted today by the multiple challenges of increasing demand for services, rising costs, and a shortage of primary manpower. Due to social forces as well as population increase, the expected number of patient visits demanded per year in dentistry will increase by 100% from 1970 to 1980 [1]. However, the net increase of dentists in active practice is expected to be only 20.3% [2]. The cost (to the dentist) of providing care has risen from an average of $3.21 per patient visit in 1962 to $8.01 per patient visit in 1971 [3,4]. This rate of increasing cost is expected to continue into the future. Clearly, if adequate care is to be provided at a cost the average citizen can afford, new patterns of dental care delivery must be implemented on a wide scale basis.

One frequently proposed solution to these problems is the introduction of expanded function auxiliaries (EFA's) into private dental practice. The EFA is trained to perform many of the routine tasks now done by the dentist but not requiring his extended formal training. Empirical studies have shown the EFA's can perform selected tasks as quickly as the dentist with no reduction in quality [5,6,7,8]. The increases in dental practice productivity (number of procedures completed per unit time) resulting from employing EFA's are reported as ranging from 92% [9] (going from two operatories, one assistant to three operatories, two assistants and one EFA) to 150% [10] (going from one operatory, one assistant to three operatories, two assistants and one EFA).

These prior studies have established the EFA as a potential solution to the problem of increasing the capacity of the dental care delivery system. However, many basic questions remain to be answered before implementation can be expected. Specific task assignments need to be determined. Also it is not clear from previous research what staffing patterns are best in given practice settings. Further, no guidelines are avaliable to permit the private practitioner to evaluate the economic and patient flow effects of introducing EFA's into his practice. Analyses indicating the tradeoffs between potential increases in patient volume and gross revenues and increases in required physical plant and personnel costs are required.

To provide adequately the answers to these questions through empirical experimentation would be inordinately time consuming and expensive; there are too many feasible combinations of staffing patterns, task assignments, facility configurations and variations in management practices which need investigation. Further, it has been observed [8] that the variation between operators in pace is frequently greater than productivity differences resulting from small variations in staffing patterns and task assignments thus further complicating empirical experiments. Also, some promising staffing patterns and task assignments are prohibited from being incorporated into field trials by existing state Dental Practice Acts.

To overcome these problems, without eliminating feasible alternatives, a computer simulation of dental practice was developed. The simulation model represents dental practice in sufficient detail so that sub-tasks can be reassigned to personnel with various assumed levels of training. Decision rules on personnel assignment are also incorporated. The model does not, however, explicitly consider the micro-motion effects of variations in instrument location, operator posture, and so forth. These effects are subsumed in the activity time distributions.

Model Development

The simulation model consists of three basic parts: the patient generator, the logical network of the treatment process, and the cost model. The patient generator produces the input patient stream according to the specifications desired for a given run of the simulator, Present input specifications include: the mean arrival rate; random, scheduled, or scheduled arrivals with random deviations from the appointment time; reason for visit mix; reason for visit sequences (scheduled input patterns); and no-show and walk-in rates. The model currently uses 98 "reason for visit" codes; a typical visit mix is shown in Table 1.

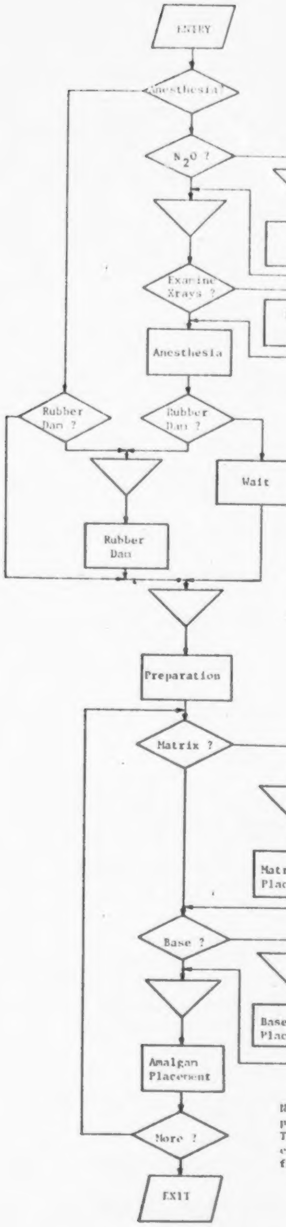
Table 1 Typical reason for visit mix

Reason for visit	Percentage	Reason for visit	Percentage
Filling	29.0	Seat Bridge	0.6
Pumice Prophylaxis, Bite Wing Stannous Fluoride	17.5	Pick Up Dentures	0.6
Check Up	10.7	Bridge Prep	0.5
Diagnosis	5.2	Lost Filling	0.5
Seat Crown	4.7	Check Crown	0.4
Extractions	4.2	Check Wisdom Teeth	0.3
Root Canal	3.2	Change Pack	0.2
Non-Coded	3.0	Inlay Prep	0.2
Toothache	2.7	Gingevectomy	0.2
Crown Prep	2.3	Seat Inlay	0.2
X-Rays	1.5	Gum Surgery	0.07
Broken Tooth	1.5	Scaling	0.07
Curretage	1.5	Check Inlay	0.07
Functional Reline	1.3	Adjust Bridge	0.07
Pumice Prophylaxis	1.1	Ream Root Canal	0.07
Impressions	1.0	Gum Treatment	0.07
Seat Denture	1.0	New Patients	3.36
Suture Removal	0.8		
Adjust Dentures	0.8		
Check Dentures	0.8		
Shade and Mould	0.7		
Check Gums	0.6		

The logical network of the treatment process dictates the flow of the patients through the dental care facility. Although the treatment pattern for any dental procedure is fixed, the task assignments and branch points within a given procedure may be altered through input parameters. Approximately 300 input parameters determine the configuration of the model with respect to: the number of personnel, the staffing pattern, the task inventories of assistants, the number of operatories, the pace rating for procedure time distributions, the physical configuration of the facility, and certain decision rules for patient management.

A portion of the logical network is shown in Figure 1. The development of this network was a critical part of the model construction. It was based on several sources. The basic structure followed that developed at the Dental Manpower Development Center in Louisville, Kentucky [8]. Also, specifications of standard dental procedures as in Bell and Grainger [11], Guralnick [12], and Kilpatrick [13] were used. Further, the networks were compared to procedures used in private dental practices located in the Southeast. Finally, dental faculty from the University of Florida, College of Dentistry and dental officers of the U.S. Public Health Service reviewed the procedure networks and made many valuable contributions. Thus, the networks employed in the final model are representative of high quality dental care as practiced with current technology in the United States.

Event time probability distributions have been determined for 135 procedure codes. A partial list of procedure codes is shown in Table 2. A typical patient will be processed through approximately a dozen procedures during any one simulated visit. Again several sources of data were employed to develop the event time probability distributions. During the five year



Note: Team changes are performed by a MACRO routine. The macro routine may be executed at any point in flow that there is a ▽.

Figure 1 Portion of Treatment Network

Table 2

Dental Procedures and Code Numbers

	ADMISSION PROCEDURES
001	Patient Preparation
002	Med. Hist. & Oral Exam.
003	Charting
004	Photography
005	Adult Radiographs
006	Pedodontic Radiographs
007	Edentulous Radiographs
008	Bitewing Radiographs
009	Single Periapical Radiographs
010	Special Radiographic Service
020	Consultation Requests
021	Oral Cytology
022	Caries Testing
023	Bleeding Time
024	Vitality Testing
025	Vitamin C Testing
026	Allergy Testing
027	Allergy Interpretation
028	Preparation of Smear
029	Smear Interpretation
050	Treatment Planning
060	Laboratory Orders
	BASIC PROCEDURES
160	Anesthesia
161	Rubber Dam Application
162	Vac-ejector Application
163	Removal of Temporary Filling
170	Tray Selection
172	Alginate Impression
175	Rubber Impression
176	Stabilizing Rubber Impression
177	Solvite Impression
	PREVENTIVE DENTISTRY
201	Oral Health Instructions
202	Scaling
203	Pumice Prophylaxis
204	Recontouring
205	Fluoride Application
206	Desensitization
207	Mouth Guard Delivery
208	Vincent's Treatment
210	Gingivectomy
211	Pack Replacement
213	Equilibration
	OPERATIVE DENTISTRY
301	Simple Amalgam Preparation
302	Compound Amalgam Preparation
303	Complex Amalgam Preparation
304	Cement Base
305	Matrix Placement
306	Amalgam Placement
311	Simple Amalgam Carved
312	Compound Amalgam Carved
313	Complex Amalgam Carved
315	Temporary Filling
321-9	One to Nine Amalgam Polished
330	Synthetic Preparation
331	Base for Synthetic
332	Synthetic Placement
341-9	One to Nine Synth. Finished
362	Compound Inlay Cavity Prep.
363	Complex Inlay Cavity Prep.
365	Direct Inlay Wax Pattern
370	Inlay Adaptation
372	Inlay Cementation & Polishing
	CROWN AND BRIDGE
401	Full Crown Preparation
402	Three-quarter Preparation
403	Jacket Preparation
404	Dowel Preparation
411	Temporary Crown
412	Temporary Shell
413	Try-in of Bridge

Louisville study [8], event times were recorded for over 250,000 observed procedures. Event time probability distributions were derived from a 100% sample of these data [14]; typical distributions are shown in Figure 2. The Louisville data, although the largest event time data bank available, suffer from the fact that they were collected in a clinical facility of unique design employing Public Health Service dentists. Thus, the Louisville data were found to be statistically different from private practice procedure times. Two steps were taken to augment the Louisville data with data more representative of private practice. First, time-lapse video tape recordings were made of 3496 procedures in a private practice in Gainesville, Florida. Second, seven private practices in Cleveland, Tennessee had ten weeks of patient visits recorded on super-8 time-lapse film. An additional 50,000 procedures were analyzed from these data. The statistical analysis of these data indicated that for most procedures the time distributions were of the same form (i.e., gamma probability distributions of order 1 to 4) but showed significantly lower mean times than the Louisville data. The final distributions used in the model were then of two types: those determined directly from private practice data; and those, for which the private practice sample sizes were insufficient, which were based on the Louisville distributions with shifts of mean. Branch probabilities within the model, and proportions of subprocedures (e.g., the proportion of filling patients receiving anesthesia) were based on private practice statistics.

An important feature of the treatment model is the classification of auxiliary personnel by skill level. In addition to a dentist, and an optional receptionist, the model can employ any mix of EFA's of any of four assumed skill levels. The duties assigned to each skill level are shown in Table 3. The assignment was based upon rankings of task difficulty determined by expert judgment [15]. This structure allows considerable flexibility in developing staffing patterns for any given simulation run. The increased knowledge assumed for the higher skill levels is translated into higher salaries in the cost model. The assumption is made that all persons possessing the requisite skill will perform a given procedure with the same quality and time distribution. This assumption has been borne out in field studies [5,6,8].

The cost model permits the evaluation of the economic consequences of alternative staffing patterns, facility configurations, and management practices. More accurately, the 'cost' model should be termed a 'net revenue' model since this is the resultant figure of merit. The model can be summarized as:

$$\begin{aligned}\text{Net Revenue} &= \text{Revenue} - \text{variable costs} - \text{fixed costs}\\ &= [\alpha\gamma'(X+Y) + S'(X+Y) + e]\\ &\quad - [NS_D + (M_C + M_L)'(X+Y) + (O+U)'(X+Y)\\ &\quad + p_t v_t + i\gamma'(X+Y)]\\ &\quad - [NT(L) - F_{p,r}(N_{B1}, C_{B1})]\end{aligned}$$

The model components are listed in Table 4. Cost and fee data are entered as input parameters for the particular practice or region being studied. Economic data have been obtained from the private practices studied, from ADA sources and from various state and federal sources.

Table 3 Procedures assigned to assistants of various skill levels

RANKED LIST OF PROCEDURES ASSIGNED TO LEVEL 1 ASSISTANTS

RANK	DESCRIPTION
1	Patient Preparation
2	Release Patient from Chair
3	Informal Introduction
4	Vac-ejector application
5	Matrix Placement
6	Amalgam Placement
7	Synthetic Placement
8	Suture Removal
9	Polish Denture
10	Rubber Dam Application

RANKED LIST OF PROCEDURES ASSIGNED TO LEVEL 2 ASSISTANT

RANK	DESCRIPTION
1-10	All Level 1 Duties
11	Charting

RANK	DESCRIPTION
12	Fluoride Application
13	Denture Patient Instruction
14	Post Surgical Instructions
15	Polish Amalgams
16	Radiography
17	Placement of Base

RANKED LIST OF PROCEDURES ASSIGNED TO LEVEL 3 & 4 ASSISTANTS

RANK	DESCRIPTION
1-17	All Level 1 & 2 Duties
18	Alginate Impression
19	Pumice Prophylaxis
20	Oral Health Instruction
21	Simple Amalgam Carved
22	Synthetic Finished
23	Compd/Complex Amalgam Carved

1-23	All Level 1, 2, & 3 Duties
24	Scaling (Incl. Subgingival)
25	Medical History and Oral Exam

Table 4 COST MODEL COMPONENTS

α -- collection ratio on fees charged

X -- vector of annual procedure frequencies by dentist (column vector)

Y -- vector of annual procedure frequencies by assistants (column vector)

γ -- fees charged for procedures (column vector)

S -- vector of externally provided fees for service (column vector)

e -- external funding not related to service

N -- number of assistants (column vector)

S_D - salary paid per assistant (row vector)

M_C - material cost per procedure (column vector)

M_L - lab costs per procedure (column vector)

O -- miscellaneous overhead per procedure (column vector)

U -- utilities cost per procedure (column vector)

v_t - fair sale value of dental care facility

P_t - local tax rate

i -- insurance rate

T(L) training and hiring cost as a function of skill level

N_{B1}- number of operatories constructed and equipped

C_{B1}- construction and equipment cost per operatory

$F_{p,r}(\cdot)$ - annual mortgage payment based on r years at p percent interest

The present version of the model is coded in GPSS/360 version 2. The model reported on here consists of over 60 functions; there are 720 blocks and two schedule routines written in PL/1, several macros and a FORTRAN help block. The GPSS language was chosen for the initial coding and model development because it is flow related and economical of programming time. Consideration is being given to recoding the fully developed model into SIMSCRIPT prior to running the full range of experiments to reduce the computer time.

Model Validation

Several approaches to validation were used. The logical validity of the treatment network was determined through judgment of a panel of University of Florida, College of Dentistry faculty. This same panel reviewed the operation of the model in detail to verify that the patient flow patterns corresponded to realistic expections. It should be noted that since the model was simulating some staffing patterns not yet available in actual practice, direct comparison between model performance and actual performance is possible over a limited range only.

The most meaningful validation would be to compare the predicted performance of a given facility - staffing pattern configuration with the performance of an equivalent actual system. Two approaches of this type are presently being pursued. A private practice, not used in the original data base, will be observed using the same video time lapse techniques referred to earlier. These data will permit the development of frequency distributions on patient waiting time, dentist optional time, the idle time of assistants, total procedures completed, patient throughput time, costs and revenue. These distributions will then be compared to those generated by the model. A shortcoming of this approach is that it tests the model against only one facility - staffing pattern configuration. Further, since EFA's are not generally available in private practice, a crucial part of the model's predictive ability will not be tested.

To overcome this problem, arrangements have been made to participate in a National Institutes of Health study [16] in which EFA's will be introduced in seven private practices with varying physical configurations. Baseline data have been collected on these same practices. Hence, the ability of the model to predict the performance changes resulting from the introduction of the EFA's will provide a good test of model validity.

Measures of Effectiveness

The development of appropriate, comprehensive measures of health systems effectiveness and cost effectiveness is always a difficult problem. An attempt should be made to balance the needs and convenience of the patient population with the needs and motivations of the primary care providers. A typical procedure is to produce multi-dimensional system performance and effectiveness measures and attempt to strike a workable balance between conflicting measures; this approach was taken here.

Measures of system performance considered were: system capacity (maximum patient arrival volume), patient waiting time, personnel utilization, gross revenue, and net revenue. No attempt was made to assign a cost to patient waiting time. Rather, this statistic was produced to allow the decision maker (whether an individual dentist or national health planner) to determine a judgmental balance between patient waiting time and other performance statistics. The net revenue figure was viewed in two ways. First, since 91% of all active dentists are in private practice [2] the net revenue figure provides a motivator to serve increasing numbers of patients when it increases as assistants are added. Second, the net revenue was set at a certain arbitrary figure (say $40,000) and the fee schedule adjusted to achieve this figure exactly. This permitted an analysis of the potential savings to the consumer population when EFA's were added but allowed the dentist to maintain a reasonable income level. Both approaches are reported in the results section below.

Experimental Results

The results reported in this section represent a small subset of all possible experiments that could be performed with this highly flexible model. The experimental phase of this project will continue through 1974. The results given here are therefore illustrative only and conclusions drawn from them should be viewed as tentative.

A typical experiment analyzes the effects of varying two factors: the number of EFA's in a four operatory facility (1, 2, 3, or 4) and the skill levels (1, 2, 3, or 4) possessed by the auxiliaries. Each realization of the sixteen resulting combinations consisted of processing 500 patients through the treatment facility. The patient arrival rate (3 patients per hour) was such that the probability of no patients being available in the waiting room was negligible except at high levels of capacity. Overflow patients were assumed lost to the system.

Table 5 indicates the summary results. As can be seen in Figure 3 the greatest marginal gain in capacity is achieved in going from level 2 to level 3. Also note that with one assistant, the level of training has no effect on productivity. This is because the assistant is tied to the dentist as a chairside assistant and can not function autonomously.

The utilization of the dentist (Figure 4) increases and the utilization of the assistants (Figure 5) decreases as the system capacity increases. The dentist is working full time to keep up with his portions of the procedures and provide necessary supervision to the assistants as the staff size increases. However, since the level 4 auxiliary can do many of the tasks formerly done by the dentist, even at high patient volumes, the dentist has more optional time with level 4 assistants than with level 1.

Gross revenues (Figure 6) follow the same pattern as the number of patient visits, as expected. Because of the low marginal gains in capacity in going from a level 3 to a level 4 auxiliary and the attendant increase in salary, the net revenue (Figure 7) picture indicates that level 3 auxiliaries dominate the level 4's. On a purely economic basis, the best staffing pattern appears to be three level 3 auxiliaries. This also provides the dentist with 42% optional time and keeps patient waiting time to reasonable levels (an average of 6 minutes from the scheduled appointment time until the patient is seated in the operatory). The computer running time for a 2000 patient experiment (including compilation) was .94 minutes on an IBM 360/65.

Table 5 Simulation Summary Results

PRIVATE OFFICE: ONE DENTIST, FOUR Rx UNITS

LEVEL OF DENTAL ASSISTANTS	#OF D.A.	PATIENT VISITS PER YEAR	DENTIST UTILIZATION	D.A. UTILIZATION	REVENUE	NET REVENUE	FEE MULTIPLIER *
1	1	2,019	.86	1.00	55.299	23.021	1.31
	2	2.419	1.00	.58	66.255	24.358	1.24
	3	2.437	1.00	.39	66.748	17.602	1.34
	4	2.437	1.00	.29	66.748	10.802	1.44
2	1	2,019	.71	1.00	55.299	22.371	1.32
	2	2.710	.96	.68	74.225	29.034	1.15
	3	2.834	1.00	.47	77.621	24.627	1.12
	4	2.795	1.00	.34	76.538	16.907	1.30
3	1	2.019	.20	1.00	55.299	21.821	1.33
	2	3.982	.42	.95	109.065	54.989	1.17
	3	5.360	.58	.88	146.807	76.830	.75
	4	5.414	.58	.68	148.286	68.009	.81
4	1	2.019	.18	1.00	55.299	20.321	1.36
	2	3.956	.38	.97	108.352	53.102	.83
	3	5.376	.53	.87	147.246	73.781	.77
	4	5.429	.50	.66	148.697	66.119	.82

*Fee schedule used in the revenue calculations must be multiplied by the factor indicated to achieve a $40,000 net revenue.

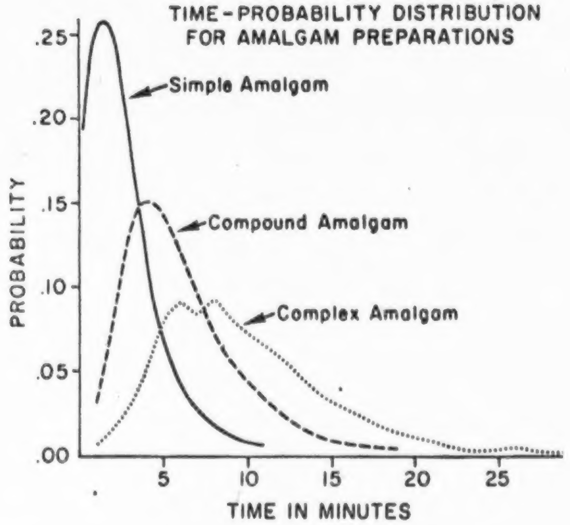


Figure 2 Probability density function for amalgam preparation.

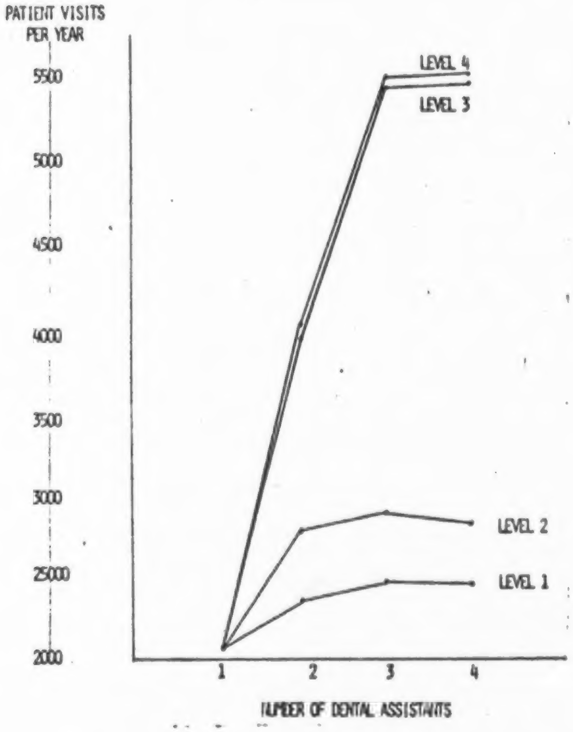


Figure 3 Productivity as a function of number of assistants.

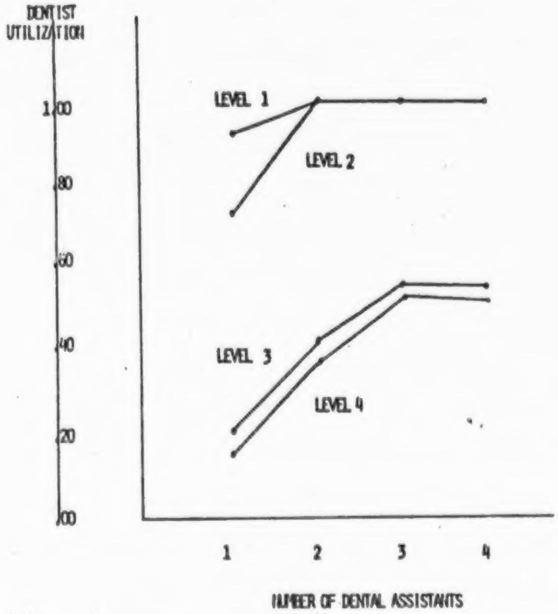


Figure 4 Dentist utilization as function of number of assistants.

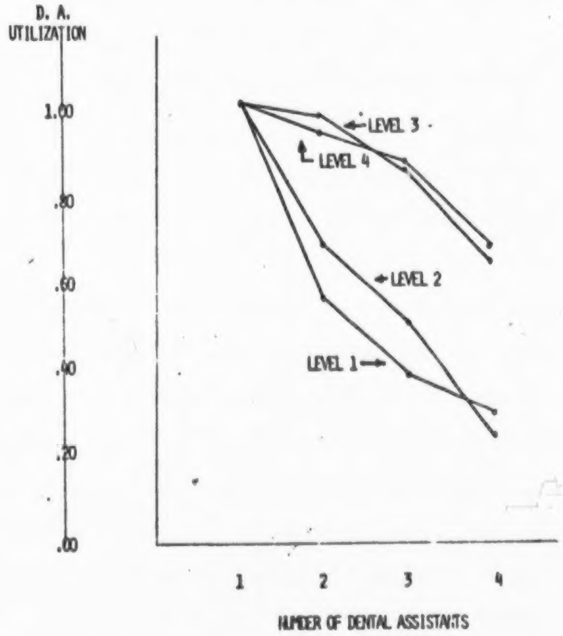


Figure 5 Dental assistant utilization as a function of number of assistants.

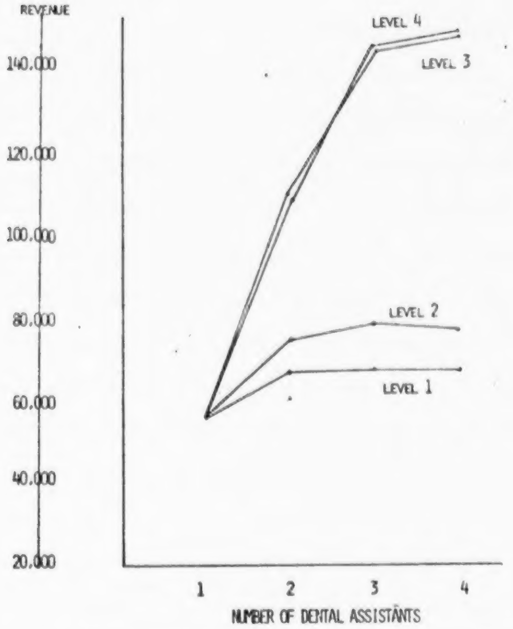


Figure 6 Revenue as a function of number of assistants.

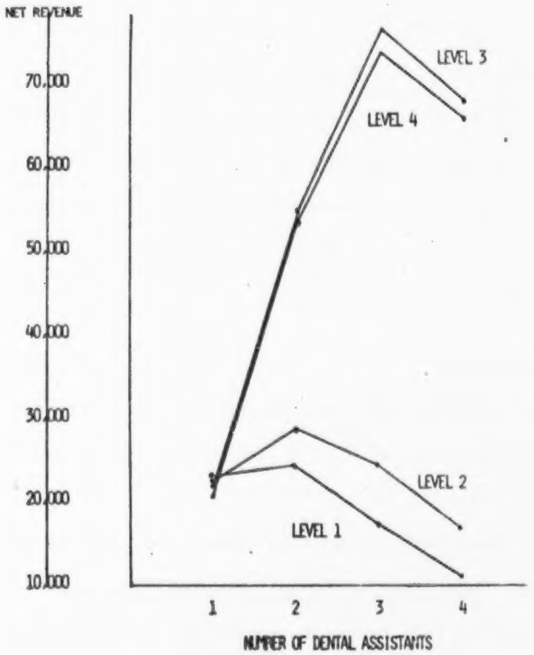


Figure 7 Net revenue as a function of number of assistants.

Conclusions and Implementation

Even in their preliminary state the foregoing results indicate that EFA's can significantly expand the capacity of a solo private practice. In addition, rather than overburdening the dentist, the proper utilization of EFA's will provide him with more optional time which can be used for continuing education, treatment planning, or other activities of direct benefit to the patient population. Further, there exits an apparent economic motivation for the dentist to use EFA's. Alternatively, the employment of EFA's offers one solution to check the rising costs of providing dental services.

With indications of such great benefits, the absence of EFA's in private practice needs some explanation. The most probable reason for their lack of use is that the American dental profession has not uet been convinced that the benefits suggested by limited field trials and simulation studies will actually accrue in practice. Further, the profession is not sure that some loss of quality of care will not occur with EFA's. Also, some dentists are uncertain as to their patients' reactions to being treated by EFA's.

The only convincing test of the model's conclusions appears to be a private practice field trial which is modeled after the 'optimum' configuration suggested by the model. As noted above, the Division of Dental Health (N.I.H.) study [16] is providing a partial demonstration of the EFA concept. Further trials of this type will however be necessary. To accomplish this goal, the developers of the simulation model are formulating plans for additional field trials which can be closely controlled and which will follow practice configuration guidelines as indicated by the simulation results.

Scope of Future Research

The collection and analysis of data, model development, validation, and preliminary experiments represent the first year's effort of a three year N.I.H. funded project. Further model development will include provision for endodontics, periodontics, and orthodontics. Provision for group practice arrangements will permit the analysis of various task distributions between specialists. Work is also progressing on incorporating improved patient scheduling procedures and team management concepts into the model to study their effects on the system performance measures. The development, test and validation steps occur on a continuing basis. The final conclusions of the research describe the exact field conditions over which a particular model configuration has been validated.

Work is also in progress to develop analytical models of the stochastic service system that is an abstraction of the dental care facility. Preliminary results [17] indicate that a steady-state GI/G/s queueing model will provide acceptable predictions of patient waiting time distributions. What is now required is the development of service distributions which are functionally dependent upon the facility configuration and staffing patterns.

The model is also being incorporated into the dental curriculum to provide a 'laboratory' for dental students to evaluate the effect of various practice management techniques on practice productivity. This on-line exposure will allow the student to observe the effects of various practice management modes over years of simulated practice in a few short sessions at a time-sharing terminal. The model will also be available to dentists already in practice for continuing education sessions.

Another ongoing study is the investigation of using the simulation model as a component of an overall planning model for national dental manpower needs. When properly validated, the model results can be extrapolated to national demand levels to predict manpower requirements under various staffing pattern mixes. The resultant manpower predictions would indicate not only the number and level of required personnel but the detailed skills inventories required for each personnel category.

References

1. American Fund for Dental Education. Meeting the challenge of the 'seventies'. JADA 82:973 May 1971.

2. Task Force on National Health Programs of the ADA. Dentistry in National Health Program. Reports of the Special Committees, ADA Chicago, October 1971.

3. A.D.A. Bureau of Research and Statistics.

Reports of councils and bureaus: 1962 survey of dental practice - summary. JADA 68:132 Jan. 1964

4. A.D.A. Bureau of Research and Statistics. Reports of councils and bureaus: 1971 survey of dental practice - summary. JADA 84:867 April, 1972.

5. Brearley, L. J. and F. N. Rosenblum. Two-year evaluation of auxiliaries trained in expanded duties. JADA 84:600 March, 1972.

6. Hammons, P. E. and H. C. Jamison. Expanded functions for dental auxiliaries. JADA 75:660 September, 1967.

7. Abromowitz, J. Expanded functions for dental assistants: a preliminary study. JADA 72:386 February, 1966.

8. Lotzkar, S., D. W. Johnson and M. B. Thompson Experimental program in expanded functions for dental assistants: phase 3 experiment with dental teams. JADA 82:1067 May, 1971.

9. Island hygienists boost productivity. J. Canad. Dent. Assn. 37:50 February, 1971.

10. Baird, K. M., E. E. Purdy and D. H. Protheroe Pilot study on advanced training and employment of auxiliary dental personnel in the Royal Canadian Corps: Final Report. J. Canad. Dent. Assn. 29:778.

11. Bell, B. H. and D. A. Grainger. Basic Operative Dentistry Procedures, second edition. Philadelphia: Lea and Febiger, 1971.

12. Guralmick, W. D., Ed. Textbook of Oral Surgery. Boston: Little, Brown, and Co., 1968.

13. Kilpatrick, H. C. Work Simplification in Dental Practice; Applied Time and Motion Studies, second edition. Philadelphia; Sanders Co., 1969.

14. Health Systems Research Division. Data Source Book for Modeling of Dental Care Delivery Systems. Working Paper DS-2-1. HSRD, University of Florida, January 1972.

15. Kingston, R. D. and T. E. Freeland, Dental Auxiliary Occupations - Task Analysis Data, Report Research and Demonstration Grant 8-0627, D.H.E.W., revised February, 1971.

16. Manpower Studies Branch. Cleveland Tennessee Productivity Study: The Effect on Private Dental Practices of Expanded Function Auxiliaries. Division of Dental Health, B.H.M.E., N.I.H., April 24, 1972.

17. del Toro, J. L. Some queueing theory models for dental practice, unpublished working paper. Health Systems Research Division, University of Florida, 1972.

THE USE OF COMPUTER SIMULATION IN HEALTH CARE FACILITY DESIGN

O. George Kennedy, Ph.D.
Manager, Management Systems - Midwestern Region
MEDICUS Systems

ABSTRACT

In this paper a decision-making tool for managers of hospitals and health care facilities, the management sciences technique of computer simulation modeling, is introduced. Its benefits as a planning and evaluative tool for hospital managers is illustrated by explaining a successful modeling application, a computer simulation model of physical therapy, developed at the Rehabilitation Institute of Chicago. With the physical therapy simulation as an example, the advantages of computer simulation modeling for containing costs and improving resource utilization and quality care are illustrated.

This paper is based on research supported, in part, by Demonstration Grant No. 12-P-55189/5-01, from the Social and Rehabilitation Service, U.S. Department of Health, Education, and Welfare, awarded to the Rehabilitation Institute of Chicago and HEW-SRS Medical Rehabilitation Research and Training Center Number Twenty.

The technique of computer simulation modeling has often been called a decision-making tool for managers. It is an especially appropriate tool for managers of health care facilities, who every day must make decisions on present operating conditions and on plans for the future, while faced with constraints on funds, staff, and facilities. Coupled with these constraints are the ever-present conditions of uncertainty, characteristic of health care services delivery, which can cause any planning effort to devolve into a guessing game.

In an effort to demonstrate that computer simulation modeling can help health care decision-making proceed on informed, objective grounds, the Rehabilitation Institute of Chicago (RIC) and the U. S. Department of Health, Education, and Welfare - Social and Rehabilitation Service Medical Rehabilitation Research and Training Center Number Twenty (RT-20) undertook a twelve-month pilot project to apply the techniques of management sciences to the delivery of rehabilitation health care services. At the onset of the study, RIC was finalizing plans for a new eighteen-story facility, now under construction, and was particularly concerned with the problems of marshalling and allocating sufficient resources to serve a projected patient load more than double its present occupancy.

An integral part of the study was the development of a computer simulation model of RIC's Physical Therapy Department -- a successful application of computer simulation modeling to be described in this paper. Benefits achieved to date from use of the model have yielded significant dollar savings and productivity increases, all without jeopardizing the quality of care delivered.

THE COMPUTER SIMULATION MODEL OF PHYSICAL THERAPY

The physical therapy department of a rehabilitation health care facility, like the one modeled here, is typically second only to the nursing service in the size of its staff and budget. It is also typically the departmental leader in producing annual revenue from patient treatment, since an overwhelming majority of the total patient population served by a rehabilitation facility will have regular physical therapy services prescribed.

The treatment delivered in a physical therapy service consists of the application of physical agents, such as heat and light, to restore gross physical function of disabled patients; continued exercise and progressive execution of simple tasks involving gross body movements and motor coordination are characteristic of the activities performed by a patient during treatment.

Although a great deal of capital equipment is utilized, physical therapy, like other areas of health care, is a labor-intensive service, drawing upon the skills of staff members who have been licensed at a number of levels of professional accreditation. It is precisely this labor-intensive characteristic of physical therapy which makes the technique of computer simulation modeling useful. In physical therapy, improvements are generally made by altering staffing patterns, not by merely rearranging work areas, altering work flows, or using more equipment. Changing staffing patterns or reallocating work loads would be an expensive, time-consuming, and potentially risky course to take in a real-world physical therapy system, if the expected outcomes of the changes were uncertain. With a computer simulation model, however, these changes can be tested before implementation, without disrupting the normal delivery of services.

Because of the importance of staffing patterns in a physical therapy service, the core of the simulation model of physical therapy described here is the allocation of manpower and facilities resources to serve any defined patient population. In the past, the computer simulation model of physical therapy has helped health care managers to make decisions on the following alternatives for staffing their physical therapy departments: 1) increasing or decreasing staff; 2) increasing the number of outpatients who can be served; 3) changing the allocation of job tasks to staff members of different skill levels; and 4) increasing services to patients of selected disability groups. Other alternatives tested for total systems operations have been: 5) changing the geographical location of physical therapy areas within a hospital; and 6) determining an equitable charging scheme for physical therapy ervices.

HOW THE MODEL WORKS

The physical therapy simulation model is a stochastic and dynamic discrete-event simulation. It accomodates the occurence of events, such as patient tardinesses and patient and therapist absences, on a random basis according to probability distributions rather than on a predetermined fixed time basis, and it accounts for the variability in time expended for treatments in the physical therapy system.

The physical therapy simulation was written in FORTRAN IV, using the FORTRAN-based simulation subroutine package SPURT/70 (Simulation Package for University Research and Teaching) developed at Northwestern University for the CDC 6400 computer system. Because of the model's basic three-component structure -- to be described below -- and the overlay and random mass storage capabilities of the CDC system, the total simulation executes in less than 43000_8 core locations and less than five minutes of computer time.

The total model is composed of three separable components: 1) a patient generating program (PATMIX); 2) the computer simulation program (PTSIM); and 3) a set of data analysis and report generating programs (RPKGS). The patient generating program, PATMIX, is used to create a number of potential patient populations to be served by the physical therapy department. The program can vary the number of patients, their disabilities, ages, sexes, and the specific physical therapy tasks prescribed for them. PTSIM, the simulation program, replicates the operations of the physical therapy system for a day, a week, or a month of time. It is through the PTSIM component that the resources of the department, such as staff and equipment, and the operating conditions of the department are altered to test alternatives. The set of data analysis and report generating programs, RPKGS, generates reports which summarize the performance of the department under various patient loads and operating conditions.

The three basic components of the model are separable, so the model may be used with either current real data from an existing physical therapy system, or with data from a simulated hypothetical operation, as indicated by the arrows in Figure I. Hypothetical patient populations may be created through PATMIX and run through the PTSIM simulation and the RPKGS data analysis programs. Or current real data from a real-world physical therapy system may be used to create a patient list for PTSIM, or for direct input to the RPKGS component only. Thus, an indepth analysis of current or planned therapy systems may be made by using all three components, and continued evaluation of physical therapy operations may be made economically by using the data analysis component only.

Model entities. The status of the physical therapy system in the model is determined through the values of its entities, both permanent and temporary.

The permanent entities in the model are physicians, therapists, aides, equipment, and other facilities. Entities are accounted for on an individual basis, except for the aides, who are treated as a manpower pool. Physician activities accounted for include only therapist assignments and activities such as medical counsel. Utilization of equipment is calculated on an hourly, daily, and weekly basis.

The temporary entities of the model are the patients and the tasks to be performed with them in physical therapy.

Model attributes. The physical therapy simulation model uses a number of demographic attributes to describe the entities of the model. The permanent attributes for therapists are therapist number, therapist name, sex, patients assigned to each therapist, physicians to whom they are assigned, scheduled times for all treatment and non-treatment tasks, hourly wages, capability for performing a number of treatment tasks, capability for treating patients with a given medical diagnosis and physical disability, maximum number of treatment hours for which they can be scheduled per day, and the maximum number of simultaneous patients and treatment tasks which each therapist can handle.

The permanent attributes for physicians are physician name, physician number, sex, therapists assigned to each physician, and diagnosis classes of patients whom each physician may treat. Permanent attributes for equipment include cost data, set-up time, usage capability, and space requirements.

The permanent attributes of patients are patient number, the therapist and physician to whom the patient is assigned, diagnosis and disability category, sex, age, therapist preference list, arrival time scheduled for therapy, admission date, list of tasks to be performed in physical therapy, list of prescribed duration times for each task to be performed, all other clinical activities scheduled for the patient for the week, and an indicator of whether the patient is an inpatient or an outpatient.

Model processes. The relationships between the entities, sets of entities, and attributes of the entities is determined by the processes involved in the functioning of the system. From Figure 2, the basic flow of patients through a physical therapy system can be seen. Starting with the beginning of a simulated working day in physical therapy, the simulation runs through all events which have been scheduled for the day. It should be noted that the model considers all patients to have been scheduled for clinical therapy on a weekly basis, i.e., patients enter physical therapy and other therapies on an appointment basis only. Randomness is introduced into the model, however, by the occurrence of unforeseen events, such as the absence of patients or scheduled therapists, the patients' arriving late for therapies, and so forth.

The recurrent events in the model, then, are the arrivals of patients for scheduled therapy visits, their performance of therapeutic tasks while in physical therapy, and their completion of tasks and departure from the physical therapy system. For therapists, a number of non-treatment tasks, such as inservice training sessions, conferences with physicians, and home visits, are recurrent events as well as are the scheduled physical therapy sessions with patients. These, and other processes and events performed in the simulation are described in more detail below.

Patient Generation Process. A patient generating program, PATMIX, is used to create a number of alternative patient populations to be served by the physical therapy department. The user supplies a matrix of physical therapy task prescriptions for patients of various disabilities, and specifies the size and disability mix of the patient population to be created. PATMIX then generates a file of patients and their demographic attributes, with a full week's schedule of rehabilitative therapies for each patient. The user is able to specify up to 75 different types of physical therapy tasks to be performed by the patients and up to 50 different disability groups to be represented in the population, all as input commands for PATMIX, and any number of files of patient population information may be generated with one run of the program.

Therapist Assignment Process. Upon the patient's entrance into the rehabilitation facility, a permanent physical therapist is assigned to care for the patient. The selection of the therapist is governed by the team concept adhered to at the facility and is strongly dependent upon therapist availability and capability. The model treats this process as already having been completed, and considers the therapist assignment to be a permanent attribute of each patient. Throughout the simulation, continuity of care is taken into consideration, with all attempts made to maintain a fixed therapist assignment for each patient.

The following processes are performed within the computer simulation program, PTSIM:

Therapy Reassignment Process. The therapy reassignment process is the crux of testing alternative staffing patterns for physical therapy. In the model, since patients are scheduled for therapy a week in advance and since patient-therapist assignments are considered permanent, if a therapist's absence is known at the beginning of the day, therapists can be assigned to treat extra patients. At the beginning of the day therapist absences are reviewed and a list is made of those patients who have to be treated temporarily by another therapist during the permanent therapist's absence. Therapist assignment is based on availability, compatibility, and capability. Multiple searches are made in the model to find a therapist who is free to take over for all the scheduled visits for the patient, under the rule that only one therapist will take over temporarily for any given patient. If no therapist is available during the patient's scheduled therapy times, searches are made through the rest of his schedule, and if possible his physical therapy treatment times are rescheduled to fit into a time when a temporary therapist will be able to work with him. Limits on the number of treatment hours for which a therapist can be scheduled, the maximum number of patients and tasks which can be handled simultaneously, and the maximum size of each therapist's daily case load are respected so that no therapist is overloaded by reassignments.

Patient Arrival Process. Patients may arrive in physical therapy either through their own accord or transported by a nurse or an aide. Since the patients are disabled, many suffering from locomotor disability, location of a previous appointment within the hospital and the distance the patient has to travel are crucial factors in determining whether or not the patient will arrive in therapy on time. A number of probability distributions in the model randomly generate patients "late", "early", or "on time" for their appointments; other routines calculate the expected travel times, based on the preceeding appointment in the patient's schedule, and generate the time the patient will arrive in therapy.

Patient Search Process. In many physical therapy departments, especially those which serve a large outpatient population, cancellation of therapy visits is frequent. In the model, therefore, a patient file is maintained for use in searching for patients who can have their therapy hours moved up to fill in times when other patients cancel. Patient rescheduling is dependent upon the patient's permanent therapist being available during the proposed rescheduled time. Reassignment of a temporary therapist is not involved in this process.

End-of-task and Discharge Process. The duration of time a patient spends in physical therapy is dependent upon his physical stamina, his desire to stay to practice techniques learned, and his remaining therapy schedule for the day. Again, using probability distributions, all of these contingencies are accomodated in the model, generating discharge times which vary from the scheduled times of departure. If the patient is due in another clinical therapy department when his scheduled time is up, the patient is discharged from physical therapy, even if not all the tasks for him were rendered. The patients are allowed to stay beyond their scheduled visit times if they do not have another appointment following physical therapy, if they are capable of performing with limited supervision, or if the regular physical therapist or appropriate supportive personnel are available to continue therapy.

When the patient leaves the department, statistical data is gathered on the equipment used, the amount of time spent by the therapist in treating the patient, the total time spent by the patient in therapy, the time spent by the patient and therapist in execution of each prescribed physical therapy task, etc. This information, along with all the permanent attributes of the patients and therapists, is written onto two event notice files, one for therapists and one for patients. An individual record for each patient visit (and for each therapist non-treatment activity) is written on the files, so that detail on the operations of the physical therapy department is retained.

MODEL OUTPUTS

The data analysis and report generating packages, RPKGS, work on the two event notice files, generating up to 17 graphical and tabular reports which have been designed for use by managers of physical therapy services and administrators of rehabilitation health care facilities. The reports depicting time variables show intervals running on five minutes, fifteen minutes, half-hourly, hourly, daily, or weekly time scales. The reports are available on a user-selection basis, and may be requested in various combinations for various time periods within a single run.

The reports focus on three common problem areas in the management of physical therapy services:

1) What is the overall allocation and utilization of the physical therapy system's resources?

2) What are the distributions of patient loads and of clinical and non-clinical activities among physicians and physical therapy staff members?

3) What are the requirements of different disability groups for physical therapy care? And what capabilities must physical therapy personnel have for treating patient populations which consist of various disability group mixes?

Figures 3, 4, and 5 concentrate on individual therapist activities. Figure 3 is a graphical display of daily therapist activities, showing the patient number of the patient treated if the therapist is engaged in patient treatment, and the name of the non-treatment task if the therapist is engaged in other business. A tabular summary of the simultaneous therapy activities performed by therapists is shown in Figure 4 which shows, also on a daily basis, the number of patients treated simultaneously and the duration of such treatments. Figure 5 is a weekly summary of the total treatment hours in which each therapist was engaged each day.

Figures 6, 7, and 8 illustrate medical staff assignments. Figure 6 is a matrix of the number of inpatients assigned to a physician-therapist team, and is designed for use in studying the team concept practiced at the facility. Figure 7 and 8 show the distribution of patients by disability group assigned to therapists and physicians, respectively. From these outputs, it is easy to see the range of capability for treating different disability groups which must be expected of active therapists and physiatrists at a rehabilitation facility; analysis may also be made of the assignement policies underlying the distribution.

Figures 9 through 11 are oriented toward individual patient analysis. Figure 9 is a summary of the number of treatment tasks performed by individual patients during a week, classified by type of treatment task. Figure 10 totals the number of treatments scheduled, received, and cancelled by each individual patient during a week. Figure 11 is a detailed summary of the distribution of arrival and departure times, both scheduled and actual, for patients each day.

Overall utilization of physical therapy facilities by patients is best illustrated by two companion outputs, Figures 12 and 13, which show, by time of day, the number of inpatients and the number of outpatients present in the physical therapy facilities. A more specific breakdown of the number of patients receiving individual treatment tasks, Figure 14, is also available. In a similar format, another report not illustrated here shows the load upon each type of therapeutic equipment used in the department. For purposes of predicting utilization and staff with different patient disability mixes, Figure 15 is generated, showing, on a weekly basis, the number of units of each therapeutic task rendered to patients in each disability group served. Variability shown in this report indicates that patient mix can, in some cases, make a significant difference in manpower and facilities utilization. Two similar reports, again not illustrated here, tabulate the breakdown of utilization according to more precise categories of "exercise" and "functional activities" tasks.

Figure 16, the summary of the therapist allocation for the total patient population, is perhaps the most useful illustration of the staff utilization patterns that are pointed out by use of the simulation model. This report tabulates the number of hours of treatment time, the number of total treatments rendered, the number and types of patients seen throughout the day, and, most importantly, the number of patients seen simultaneously and the number of different treatment tasks supervised simultaneously. As can be seen from the example of this output shown in Figure 16, at times the real-world allocation of workloads in a physical therapy department is far from equitable.

RESULTS AND CONCLUSIONS

Although developed during a pilot research and demonstration project, the use of the physical therapy simulation model has had some very pragmatic cost-saving results for the rehabilitation facility modeled. Concretely, recommendations of the simulation study have led to the following improvements in the delivery of physical therapy services:

1) It was found that 70% of the treatment delivered in the department could be rendered from decentralized areas closer to the patient's rooms. Putting an exercise area on each patient ward and eliminating one floor of planned central physical therapy space, as recommended, will save a total of $576,000 in construction costs for a new 18-story, $26 million rehabilitation facility being built.

2) An estimated additional $10,000 increase in revenue annually will be obtained by decreasing patient travel time to the decentralized physical therapy areas and thereby reducing patient tardinesses and absences.

3) The physical therapy staff could be reduced by one-third, at an annual savings of $50,000. To date, with staff positions frozen, normal staff attrition has reduced the physical

therapy staff expenses by $20,000.

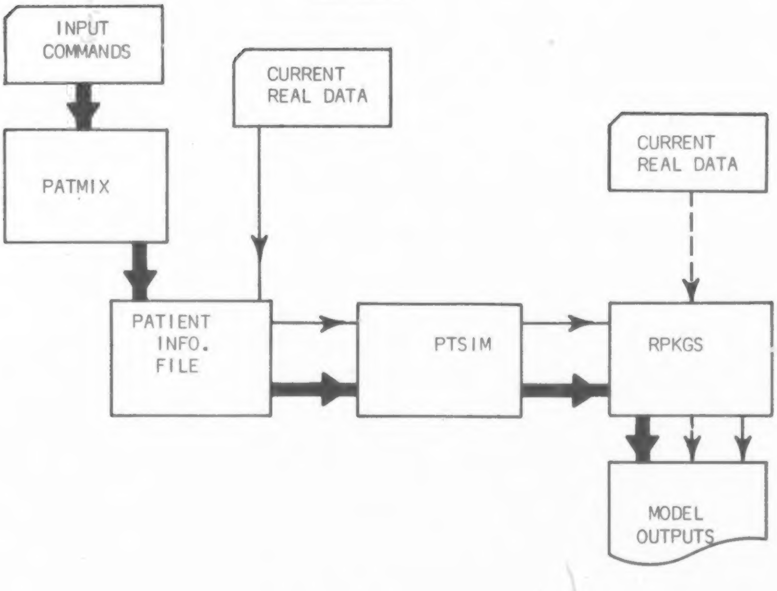
4) Even with decreased staff, the productivity of the physical therapy department has increased due to better scheduling of patient treatments and improved assignment of tasks to staff members of various skill levels. To date, productivity has increased by 30%, and the capacity to treat a burgeoning outpatient load has been increased by 25%.

More important for future research and progress in health care operations, however, are some results which cannot be so easily quantified. The first is the benefit shown by improved scheduling and rescheduling of patient treatments in physical therapy. In such a service, the performance of the total system may hinge upon the implicit rules used in assigning treatment times to incoming patients. Through use of the model, it has been demonstrated to physical therapy personnel that a relatively small number of patient and therapist attributes may be used to match capability, availability, and compatibility during a rescheduling effort. Hopefully, professionals in the field will work to define other of the "human factors" operative in scheduling, and advanced technological capabilities may be utilized in the future to perform much of the scheduling effort, to the benefit of overall departmental performance.

A second result is the awareness that the modeling approach is one step in proving that a large number of hospitals, although they savor their professed "uniqueness", are in fact very much alike. For instance, the variability of attributes, treatment tasks, staff and patient population sizes, etc., in the physical therapy model developed here have been used to show that, regardless of the specifics of the operating environment, the basic flow of activities through one physical therapy department resembles that of virtually any other physical therapy department. It is hoped that this characteristic of the modeling technique will be used in the future to produce results that may have "industry-wide" impact on our nation's health care system.

BASIC STRUCTURE
OF MODEL

FIGURE I


INPUT COMMANDS
CURRENT REAL DATA
PATMIX
CURRENT REAL DATA
PATIENT INFO. FILE
PTSIM
RPKGS
MODEL OUTPUTS

200

GENERAL PRIMARY SYSTEMS FLOW

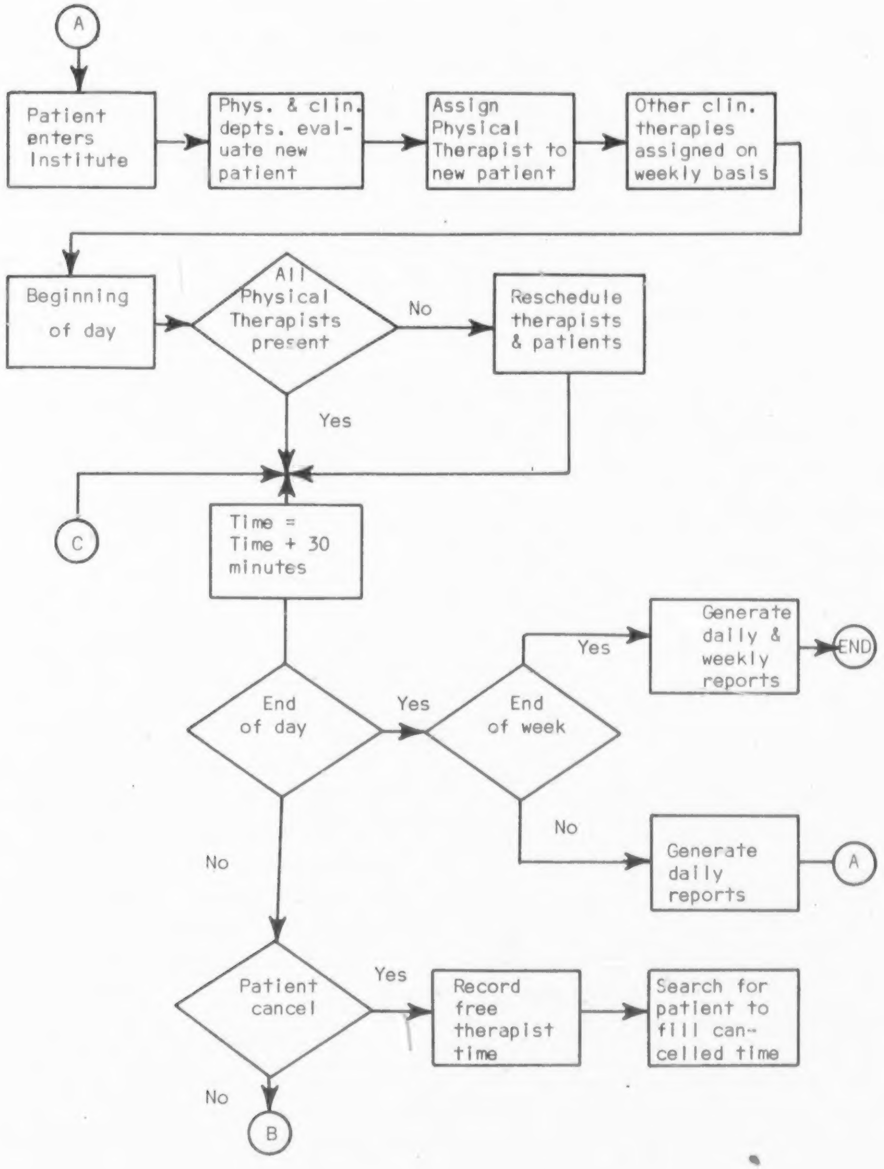

A
Patient enters Institute
Phys. & clin. depts. evaluate new patient
Assign Physical Therapist to new patient
Other clin. therapies assigned on weekly basis
Beginning of day
All Physical Therapists present
No
Reschedule therapists & patients
Yes
C
Time = Time + 30 minutes
Generate daily & weekly reports
Yes
END
End of day
Yes
End of week
No
Generate daily reports
A
No
Patient cancel
Yes
Record free therapist time
Search for patient to fill cancelled time
No
B

FIGURE 2

GENERAL PRIMARY SYSTEMS FLOW (con't)

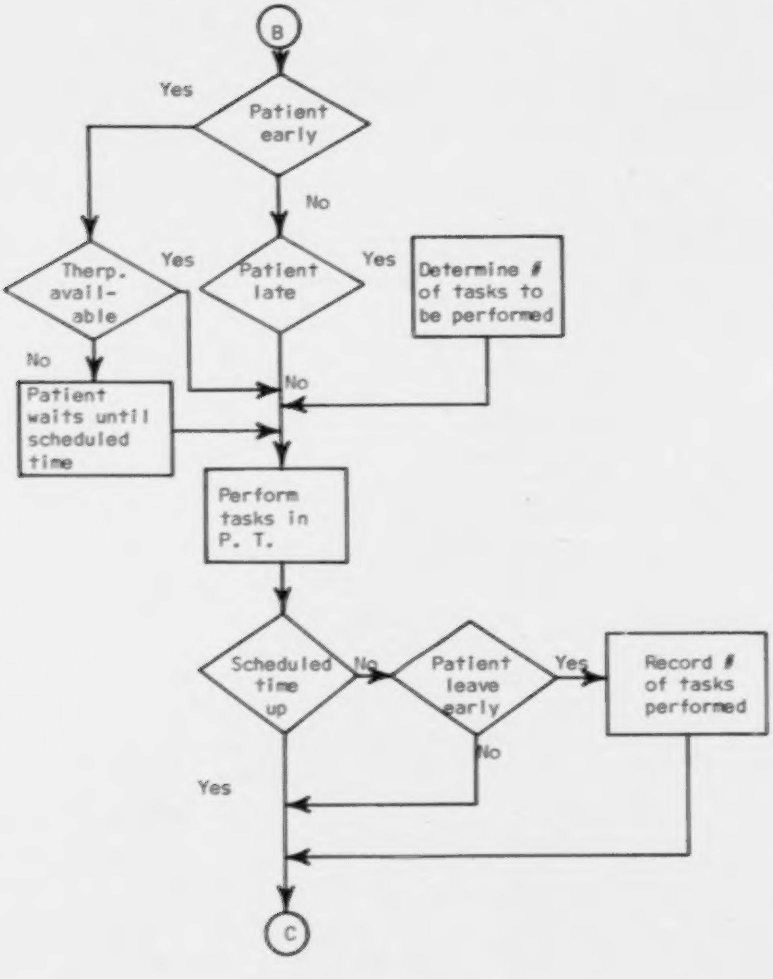
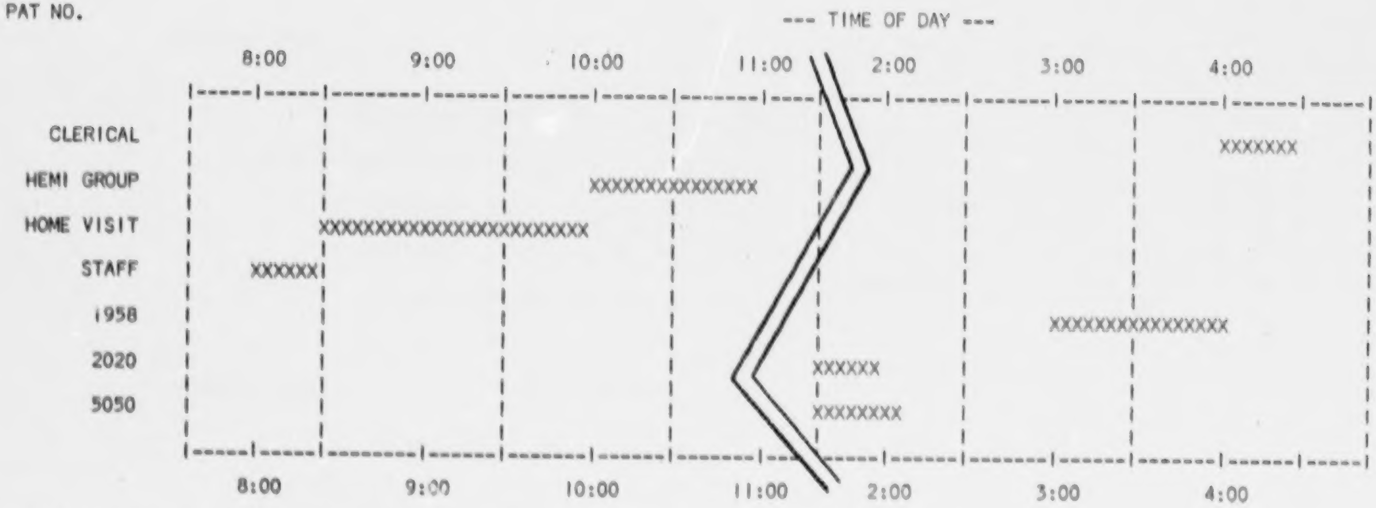

B
Patient early
Yes
No
Therp. available
Yes
No
Patient late
Yes
No
Determine # of tasks to be performed
Patient waits until scheduled time
Perform tasks in P. T.
Scheduled time up
No
Yes
Patient leave early
Yes
No
Record # of tasks performed
C

FIGURE 2

PHYSICAL THERAPY DEPARTMENT

DAILY THERAPIST ANALYSIS

THERAPIST NAME--- THERAPIST NO. DATE--- / /



NOTE: Figure is only partial output of computer print-out.

FIGURE 3

THERAPIST TREATMENT SUMMARY

```
PHYSICAL THERAPY DEPARTMENT

                    THERAPIST TREATMENT SUMMARY

THERAPIST NAME             THERAPIST NO.                   DATE --   /   /


                 0 PATIENTS TREATED DURING 1.50 HOURS

                 1 PATIENTS TREATED DURING 1.00 HOURS

                 2 PATIENTS TREATED DURING 3.25 HOURS

                 3 PATIENTS TREATED DURING 1.25 HOURS

                 4 PATIENTS TREATED DURING 1.00 HOURS

THERAPIST NAME             THERAPIST NO.                   DATE --   /   /


                 0 PATIENTS TREATED DURING 1.75 HOURS

                 1 PATIENTS TREATED DURING 4.75 HOURS

                 2 PATIENTS TREATED DURING 1.50 HOURS
```

FIGURE 4

PHYSICAL THERAPY DEPARTMENT

THERAPIST ATTENDANCE SUMMARY WEEK OF / /

THERAPIST	3/18	3/19	3/20	3/21	3/22	3/23	3/24
	7.00	6.75	0.00	0.00	6.25	6.25	6.25
	7.00	7.50	0.00	0.00	7.50	7.00	7.25
	6.25	5.75	0.00	0.00	6.50	6.50	6.00
	7.00	7.25	0.00	0.00	4.50	6.50	6.00
	3.25	7.00	0.00	0.00	6.25	6.75	7.25
	6.50	5.00	0.00	0.00	6.25	7.00	6.25
	4.00	4.25	0.00	0.00	3.50	3.00	4.00
	6.00	6.00	0.00	0.00	4.50	6.25	5.50
	7.00	7.00	0.00	0.00	6.75	6.00	6.50
	6.25	6.50	0.00	0.00	6.50	6.25	7.00
	3.75	5.25	0.00	0.00	5.25	2.00	5.00
	6.00	6.50	0.00	0.00	5.50	5.25	6.50
	7.25	3.75	0.00	0.00	6.00	5.00	5.25
	4.25	3.50	0.00	0.00	3.50	4.25	3.75

FIGURE 5

```
PHYSICAL THERAPY DEPARTMENT


PHYSICIAN-THERAPIST ASSIGNMENT            WEEK OF    /  /

                           --- PHYSICIAN NUMBER ---
                            7     11     50     65     80

THERAPIST NO.                                                  TOTALS

       7                    1      1      1      2      1         6
       9                    1      0      4      0      1         6
      13                    2      0      1      1      1         5
      17                    2      2      2      0      0         6
      20                    1      1      1      2      2         7
      54                    0      1      0      2      4         7
      60                    1      0      0      0      1         2
      65                    3      0      0      1      1         5
      70                    2      0      1      2      3         8
      72                    2      0      1      1      0         4
      78                    0      3      0      0      0         3
      80                    1      0      1      3      2         7
      88                    0      1      0      1      1         3
      98                    0      2      2      0      0         4

   TOTALS                  16     11     14     15     17        73

LEGEND:  Matrix represents number of in-patients assigned
         to a physician and a therapist.
```

FIGURE 6

```
PHYSICAL THERAPY DEPARTMENT

DIAGNOSIS VS. TASKS RENDERED SUMMARY              WEEK OF   / /

                             --- TASKS ---
      DIAGNOSIS              B  FX     G   H    TR LET   F   ROM  MT  XTRA  TOTALS
      ---------
                           ---------------------------------------------------------
A/K AMPUTEE                  1  13    11   0     2  2    2     0   0    0      49
BILATERAL A/K AMP            8  19     9   0     0 19    0     0   0    0      55
BILATERAL B/K AMP            0   0     0   0     0  9    0     0   0    0       9
OTHER OR MULT. AMP           0   9     2   2     0 10    0     1   0    0      26

TOTAL AMPUTEES               9  41    22   2     2 58          1   0    0     139


ARTHRITIS                    0  18    11  10     0  0    0     2   0    0      50
RHEUMATOID ARTHRITIS         0  11     6   6     0  0    3     1   0    0      36

TOTAL ARTHRITIS              0  29    17  16     0  0    3         0    0      86


BURN                         0   8     0   0     0  0    0     0   0    4      19
CEREBRAL PALSY               0   7     9   0     0  0    5         0    1      30
C.V.A                        0   3     4   0     0  0    2     0   0    5      18
EVALUATION OR EMG            0   3     4   0     0  0    0     0   1    4      13


LEFT HEMIPLEGIA              0  24    22   0     0  0    1     4   0   11      85
RIGHT HEMIPLEGIA             0   8     4   3     0  0    0     0   0    0      15
RIGHT HEMI/APHASIA           0  21    21   0     0  0   10     0   0    3      64

TOTAL HEMIPLEGIA             0  53    47   3     0      24     4   0   14     164


POST FRACTURE                0  12    11   0     0  0    6     2   0    0      40
POST POLIO                   0   5     0   0     0  0    3     0   0    0       8


PARAPLEGIA/PARESIS           1 119    58   0     0  0   45    12   2   26     303
QUADRIPLEGIA/PARESIS         0  53    15   0     0      31     5   0    0     108

TOTAL SPINAL CORD            1 172    73   0     0  0   76    17   2   26     411


POST SURGERY                 0   5     5   0     0  0          0   0    4      19
OTHER INJURY                 0  13     9   0     0  0          0   0    0      34
OTHER PARALYSIS              0  17     3   1     0  0    8     3   3    0      43
CENTRAL NERVOUS              0   9     8   0     0  0    9     0   0    0      27
CARDIO-VASCULAR              1   5     5   0     0  0    0     3   0    0      16
MUSCULAR SKELETAL            0   3     6   0     0  7    3     0   0    0      22

    ---   GRAND TOTALS      11 385   223  22     2 65  147     3   6   58    1089

                                                NOTE:  Figure is only partial
                                                       output of computer
                                                       print-out.
```

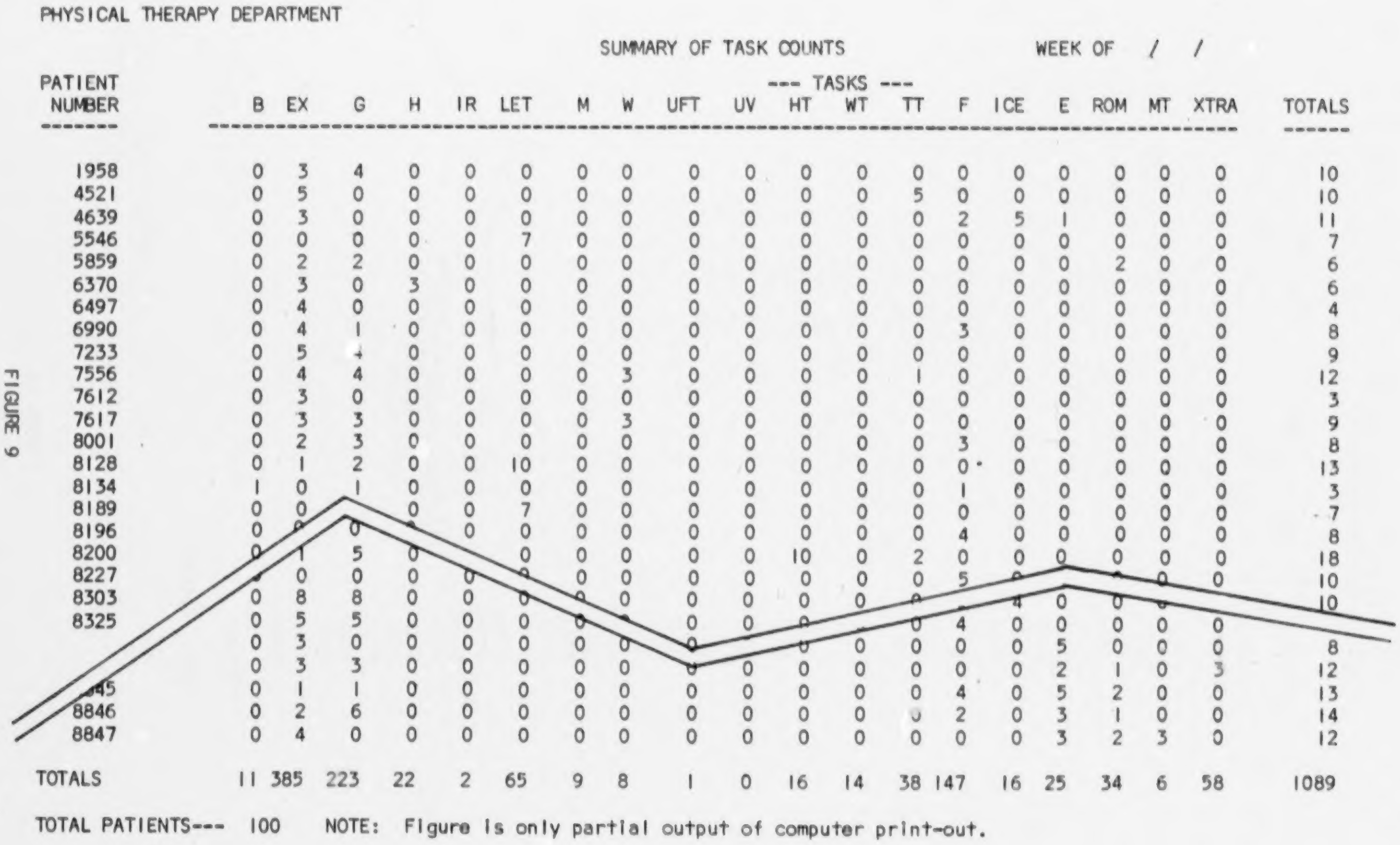
FIGURE 7

PHYSICAL THERAPY DEPARTMENT

MEDICAL STAFF ASSIGNMENT VS. DIAGNOSIS CLASS WEEK OF / /

	--- PHYSICIAN NUMBER ---					
DIAGNOSIS	7	11	50	65	80	TOTALS
A/K AMPUTEE	0	0	1	2	0	3
BILATERAL A/K AMP	1	0	0	1	0	2
BILATERAL B/K AMP	1	0	0	0	0	1
OTHER OR MULT. AMP	0	0	0	1	0	1
TOTAL AMPUTEES	2	0	1	4	0	7
ARTHRITIS	1	0	0	1	0	2
RHEUMATOID ARTHRITIS	0	1	0	1	0	2
TOTAL ARTHRITIS	1	1	0	2	0	4
BURN	0	1	0	0	0	1
CEREBRAL PALSY	0	3	0	0	0	3
C.V.A.	1	0	0	0	0	1
EVALUATION OR EMG	0	0	0	1	0	1
LEFT HEMIPLEGIA	3	0	2	3	4	12
RIGHT HEMIPLEGIA	0	0	0	0	1	1
RIGHT HEMI/APHASIS	2	0	2	0	1	5
TOTAL HEMIPLEGIA	3	0	2	3	4	12
POST FRACTURE	1	0	0	0	1	2
POST POLIO	0	0	1	0	0	1
PARAPLEGIA/PARESIS	6	3	5	4	5	23
QUADRIPLEGIA/PARESIS	2	0	4	0	3	9
TOTAL SPINAL CORD	8	3	9	4	8	32
POST SURGERY	0	0	0	0	1	1
OTHER INJURY	0	2	0	0	1	3
OTHER PARALYSIS	0	0	1	0	2	3
CENTRAL NERVOUS	0	1	0	1	0	2
--- GRAND TOTALS	16	11	14	15	17	73

FIGURE 8



```
PHYSICAL THERAPY DEPARTMENT

                                      SUMMARY OF TASK COUNTS          WEEK OF  /  /
PATIENT                                          --- TASKS ---
 NUMBER      B  EX   G   H  IR LET   M   W UFT  UV  HT  WT  TT   F ICE   E ROM  MT XTRA   TOTALS
-------     -------------------------------------------------------------------------   ------

  1958       0   3   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0       10
  4521       0   5   0   0   0   0   0   0   0   0   0   0   5   0   0   0   0   0   0       10
  4639       0   3   0   0   0   0   0   0   0   0   0   0   0   2   5   1   0   0   0       11
  5546       0   0   0   0   0   7   0   0   0   0   0   0   0   0   0   0   0   0   0        7
  5859       0   2   2   0   0   0   0   0   0   0   0   0   0   0   0   0   2   0   0        6
  6370       0   3   0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0        6
  6497       0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0        4
  6990       0   4   1   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0   0        8
  7233       0   5   4   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0        9
  7556       0   4   4   0   0   0   0   3   0   0   0   0   1   0   0   0   0   0   0       12
  7612       0   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0        3
  7617       0   3   3   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0        9
  8001       0   2   3   0   0   0   0   0   0   0   0   0   0   3   0   0   0   0   0        8
  8128       0   1   2   0   0  10   0   0   0   0   0   0   0   0   0   0   0   0   0       13
  8134       1   0   1   0   0   0   0   0   0   0   0   0   0   1   0   0   0   0   0        3
  8189       0   0       0   0   7   0   0   0   0   0   0   0   0   0   0   0   0   0        7
  8196       0   0   0   0   0   0   0   0   0   0   0   0   0   4   0   0   0   0   0        8
  8200       0   1   5   0       0   0   0   0   0  10   0   2   0   0   0   0   0   0       18
  8227       0   0   0   0   0   0   0   0   0   0   0   0   0   5   0       0   0   0       10
  8303       0   8   8   0   0   0   0   0   0   0   0   0   0       4   0   0   0           10
  8325       0   5   5   0   0   0   0   0   0   0   0       0   4   0   0   0   0   0
             0   3   0   0   0   0   0   0   0       0   0   0   0   0   5   0   0   0        8
             0   3   3   0   0   0   0   0   0   0   0   0   0   0   0   2   1   0   3       12
    45       0   1   1   0   0   0   0   0   0   0   0   0   0   4   0   5   2   0   0       13
  8846       0   2   6   0   0   0   0   0   0   0   0   0   0   2   0   3   1   0   0       14
  8847       0   4   0   0   0   0   0   0   0   0   0   0   0   0   0   3   2   3   0       12

TOTALS      11 385 223  22   2  65   9   8   1   0  16  14  38 147  16  25  34   6  58     1089

TOTAL PATIENTS---  100    NOTE:  Figure is only partial output of computer print-out.
```

FIGURE 9

PHYSICAL THERAPY DEPARTMENT

PATIENT ATTENDANCE SUMMARY WEEK OF / /

PATIENT NUMBER	VISITS SCHEDULED	VISITS RENDERED	VISITS CANCELLED
1958	8	7	1
4521	5	5	0
4639	5	5	0
5546	10	7	3
5859	2	2	0
6370	3	3	0
6497	5	4	1
6990	5	4	1
7233	5	5	0
7556	5	5	0
7612	5	3	2
7617	3	3	0
8001	3	3	0
8128	10	10	0
8134	1	1	0
8189	8	7	1
8196	5	4	1
8200	10	10	0
8227	5	5	0
8303	5	5	0
8325	10	9	1
8336	4	2	2
8357	4	4	0
8376	1	1	0
8421	3	3	0
8440	10	10	0
8446	5	5	0
8487	9	8	1
8513	2	2	[illegible]
8519	3	3	0
8525	5	4	0
[illegible]	10	9	0
8828	10	[illegible]	0
8832	5	8	2
8834		5	0
8837	5	5	0
8842	5	5	0
8845	8	6	2
8846	8	7	1
8847	4	4	0
8848	1	0	1
TOTALS	620	564	56

TOTAL PATIENTS 101

NOTE: Figure is only partial output of computer print-out.

FIGURE 10

```
PHYSICAL THERAPY DEPARTMENT

     PATIENT THERAPY SCHEDULE                                  DATE ---    /  /

                         MORNING                                  AFTERNOON
          -----------------------------------     ---------------------------------

              SCHEDULED           ACTUAL              SCHEDULED          ACTUAL
PATIENT
 NUMBER       IN     OUT        IN     OUT            IN    OUT          IN    OUT

   1958    10:00   11:00     10:00   11:00         12:30   1:30       12:30   1:50
   4521                                             2:00   3:00        2:00   3:00
   4639                                             2:30   3:30        2:30   3:30
   5546     9:00   10:00     -----   -----          1:00   3:00        1:00   3:00
   6370     9:30   10:30      9:30   10:30
   6497                                             2:30   3:30        2:30   3:30
   6990    10:30   11:30     10:30   11:30
   7233                                             3:00   4:00        3:00   4:00
   7556    11:00   12:00     11:15   12:00
   7612                                             1:00   2:00        1:00   2:00
   7617     9:00   11:00      9:00   11:00
   8001     8:30    9:30      8:30    9:30
   8128     9:00   11:30      9:00   11:30         12:30   3:30        1:00   3:30
   8189     9:00   11:30     -----   -----          1:00   2:00        1:00   2:00
   8196                                             1:30   2:30        1:30   2:30
   8200    10:00   11:00     10:00   11:10          2:30   3:30        2:30   3:50
   8227                                             2:30   3:30        2:30   3:30
   8303                                             2:00   3:00        2:00   3:00
   8325     9:00   10:30      9:00   10:30          1:00   2:30        1:00   3:00
   8357    10:00   11:30     10:00   12:00          1:30   3:00        1:30   3:25
   8421    10:00   11:00     10:00   11:00
   8440     8:30   10:00      8:30   10:00          1:00   3:30        1:00   3:50
   8446     9:00   10:00      9:00   10:00
   8487     8:30    9:30     -----   -----
   8513                                             1:00   3:00        1:00   3:00
   8519                                             2:00   3:00        2:00   3:00
   8525    10:00   11:00     10:00   11:00
   8528     8:30    9:       -----   -----         12:30   1:30        1
   8537     8:                8:30    9:45          1:00   2:00
   85              9:30                             2:00               1:00   3:15
                                     10:30                 3:14        -----  3:;4
   8797    10:00   12:00     10:00
   8812                                             1:30   2:30        1:30   2:30
   8816     8:30    9:30      8:30    9:30
   8823     8:30   10:00      8:30   10:00
   8826                                             2:00   3:00        2:00   3:00
   8832    10:00   11:00     10:00   11:00         12:30   1:30       12:30   1:30
   8834    10:00   11:00     10:00   11:45
   8837     9:30   10:30      9:30   10:40
   8842    10:00   11:00     10:30   11:25
   8845    10:00   11:00     10:00   11:00          1:00   2:00        1:25   2:00
   8846     8:30   10:30      8:30   10:30          3:00   4:00        -----  -----
   8847                                             2:30   3:30        2:30   3:30

TOTAL PATIENTS:  89
                                                NOTE:  Figure is only partial output
NOTE:  CANCELLATION DENOTED BY:  -----                 of computer print-out.
```

FIGURE II

NO. OF IN-PATIENTS IN PHYSICAL THERAPY VS. TIME OF DAY

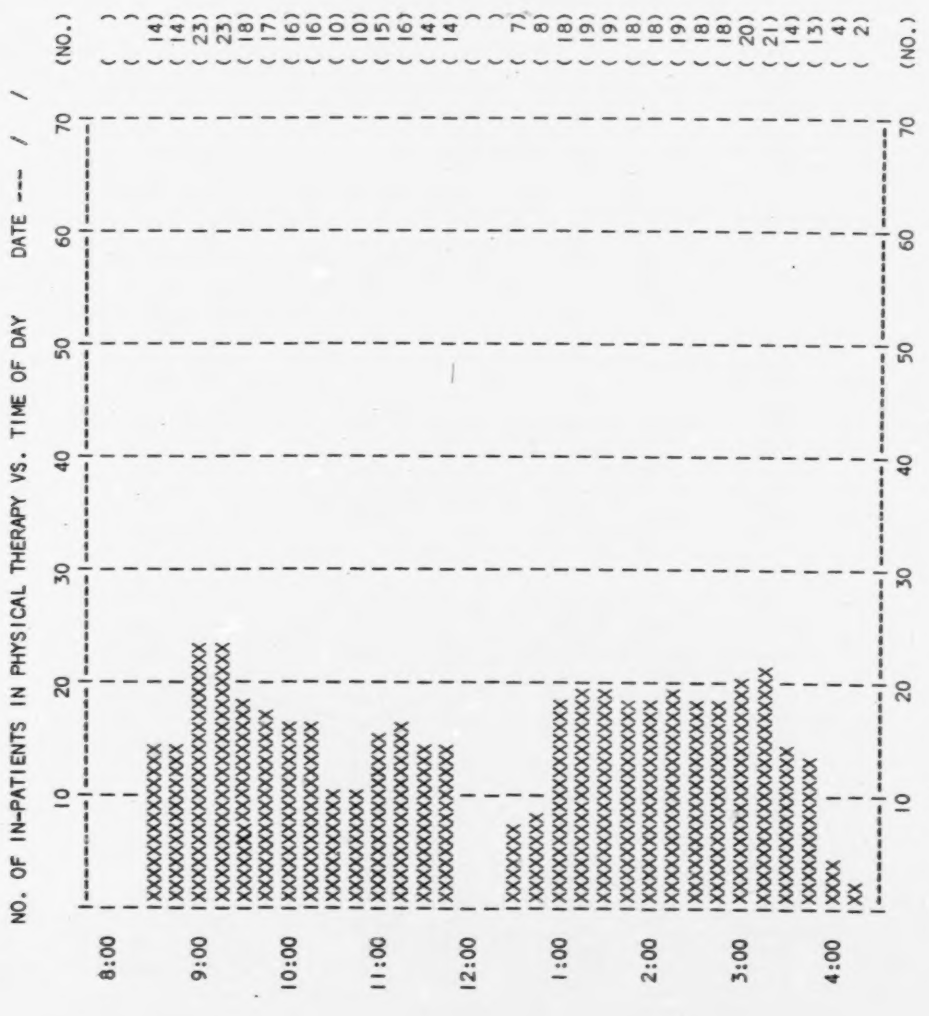


FIGURE 12

NO. OF OUT-PATIENTS IN PHYSICAL THERAPY VS. TIME OF DAY

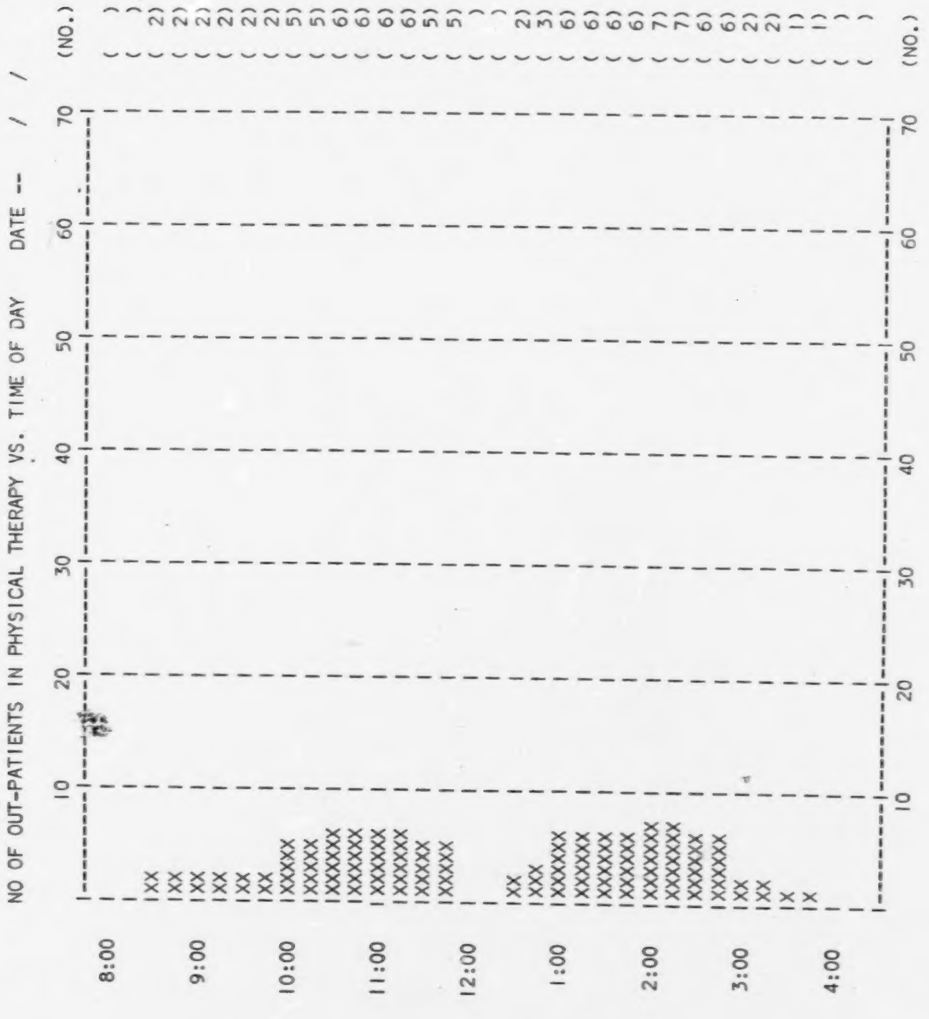


FIGURE 13

```
PHYSICAL THERAPY DEPARTMENT

          NO. OF PATIENTS RECEIVING TREATMENT VS. TIME OF DAY

               TASK:  EXERCISE              DATE --   /   /


                   10         20         30         40         50    (NO.)
         I----------I----------I----------I----------I----------I
  8:00   I          I          I          I          I          I    (   )
         I          I          I          I          I          I    (   )
         IXXXXXXXX  I          I          I          I          I    (  8)
         IXXXXXXXXXI-          I          I          I          I    (  9)
  9:00   IXXXXXXX   I          I          I          I          I    (  7)
         IXXXXXXX   I          I          I          I          I    (  7)
         IXXXXXXXXXX           I          I          I          I    ( 10)
         IXXXXXXXXXXX          I          I          I          I    ( 11)
 10:00   IXXXXXX    I          I          I          I          I    (  6)
         IXXXXXX    I          I          I          I          I    (  6)
         IXXXX      I          I          I          I          I    (  4)
         IXXXXX     I          I          I          I          I    (  5)
 11:00   IXXXXXXXXXI           I          I          I          I    (  9)
         IXXXXXXXX  I          I          I          I          I    (  8)
         IXXXXX     I          I          I          I          I    (  5)
         IXXXXXX    I          I          I          I          I    (  6)
 12:00   I          I          I          I          I          I    (   )
         I          I          I          I          I          I    (   )
         IXXXX      I          I          I          I          I    (  4)
         IXXXX      I          I          I          I          I    (  4)
  1:00   IXXXXXXXXXX           I          I          I          I    ( 10)
         IXXXXXXXXXX           I          I          I          I    ( 10)
         IXXXXXXX   I          I          I          I          I    (  7)
         IXXXX      I          I          I          I          I    (  4)
  2:00   IXXXXX     I          I          I          I          I    (  5)
         IXXXX      I          I          I          I          I    (  4)
         IXXXXX     I          I          I          I          I    (  5)
         IXXXXXX    I          I          I          I          I    (  6)
  3:00   IXX        I          I          I          I          I    (  2)
         IXXX       I          I          I          I          I    (  3)
         IXXX       I          I          I          I          I    (  3)
         IXX        I          I          I          I          I    (  2)
  4:00   I          I          I          I          I          I    (   )
         I          I          I          I          I          I    (   )
         I          I          I          I          I          I    (   )
         I          I          I          I          I          I    (   )
  5:00   I          I          I          I          I          I    (   )
         I----------I----------I----------I----------I----------I
                   10         20         30         40         50    (NO.)
```

FIGURE 14

```
PHYSICAL THERAPY DEPARTMENT

MEDICAL STAFF ASSIGNMENT VS. DIAGNOSIS CLASS     WEEK OF    /   /

                                 --- THERAPIST NUMBER ---

DIAGNOSIS                   7      9     13     80     88     98   TOTALS

A/K AMPUTEE                 2      0      0      0      0      0      3
BILATERAL A/K AMP           1      0      0      0      0      0      2
BILATERAL B/K AMP           0      0      0      1      0      0      1
OTHER OR MULT. AMP          0      0             0      0      0      1

TOTAL AMPUTEES              3      0      0      1      0      0      7


ARTHRITIS                   0      0      0      1      0      0      2
RHEUMATOID ARTHRITIS        0      0      0      1      0      0      2

TOTAL ARTHRITIS             0      0      0      2      0      0      4


BURN                        0      0             0      0      0      1
CEREBRAL PALSY              0      0      0      0      1      0      3
C.V.A.                      0      1      0      0      0      0      1
EVALUATION OR EMG           0      0      1      0      0      0      1


LEFT HEMIPLEGIA             1      0      1      0      0      0      6
RIGHT HEMIPLEGIA            0      0      0      0      0      0      1
RIGHT HEMI/APHASIA          0      0      1      1      0      0      5

TOTAL HEMIPLEGIA            1      0             1      0      0     12


POST FRACTURE               0      0      0      0      0      0      2
POST POLIO                  0      1      0      0      0      0      1


PARAPLEGIA/PARESIS          1      2      0      2      1      2     23
QUADRIPLEGIA/PARESIS        0      1      0      1      0      1      9

TOTAL SPINAL CORD           1      3      0      3      1      3     32


POST SURGERY                0      0      0      0      1      0      1
OTHER INJURY                1      0      0      0      0      0      3
OTHER PARALYSIS             0      1      0      0      0      0      3
CENTRAL NERVOUS             0      0      0      0      0      1      2

    --- GRAND TOTALS        6      6      5      7      3      4     73
```

LEGEND: Matrix represents number of in-patients assigned to a therapist.
Figure is only partial output of computer print-out.

FIGURE 15

THERAPIST ALLOCATION FOR 14 THERAPISTS
TREATING PATIENT POPULATION: 71 IN-PATIENTS, 17 OUT-PATIENTS

THERAPIST NUMBER	TOTAL # OF TREATMENT HOURS	MAX. # OF PATIENTS TREATED*	MAX # OF TASKS PERFORMED*	# OF PATIENTS WITH ONE VISIT	# OF PATIENTS WITH TWO VISITS	TOTAL PATIENTS	TOTAL VISITS
1	6.5	5	5	3	5	8	13
2	6.5	4	3	6	2	8	10
3	4.5	3	2	5	1	6	7
4	6.0	2	2	8	0	8	8
5	6.0	4	3	8	0	8	8
6	6.5	3	3	4	3	7	10
7	4.0	2	2	0	2	2	4
8	6.0	3	2	2	3	5	8
9	6.5	3	3	7	2	9	11
10	6.0	4	4	5	2	7	9
11	4.0	3	3	2	2	4	6
12	6.5	3	3	4	4	8	12
13	5.0	2	1	3	1	4	5
14	2.5	3	2	4	0	4	4

*Treatment based on 15 minute intervals
(Therapists #7 and #11 are supervisors, maximum of 4.5 treatment hours per day)

FIGURE 16

A SIMULATION MODEL OF A UNIVERSITY HEALTH SERVICE OUTPATIENT CLINIC

by

Robert Baron
Research Assistant, Department of Industrial Engineering and Operations Research
University of Massachusetts
Amherst, Massachusetts

and

Edward J. Rising
Professor of Industrial Engineering and Operations Research
University of Massachusetts
Amherst, Massachusetts

October, 1972

Sixth Annual Joint Simulation Conference
San Francisco, California (January 17-19, 1973)

Abstract

This paper focuses on the development of a simulation model of a University Health Service Outpatient Clinic the implementation of which has resulted in significant improvements to system performance. The details of these improvements are published elsewhere; they amouted to savings in excess of fifty thousand dollars the first year the model was used, improved physician morale, and acceptance on the part of the Health Service staff of the simulation model as a tool for decision-making.

The Health Service provides complete outpatient medical care and limited inpatient care for about 19,000 people. The resulting outpatient load of over 400 persons a day requires the services of 12 full-time physicians. The simulation model for which appointment and walk-in patients are generated separately, was developed over a two-year period and takes the general form of a multiple stage, parallel queueing system with a variable number of servers. Validation problems are discussed, and data is presented.

Introduction

The literature in the field of health administration reflects the increasing importance of and burgeoning national interest in the delivery of health care through outpatient facilities. This paper reports on the details of a simulation model that was used as a portion of an overall systems analysis made of the delivery of outpatient care at the University of Massachusetts Health Service. The manner in which this systems analysis was carried out and the results it achieved were described in the Journal of the American College Health Association in a three article series in the June 1972 issue (1, 2, 3). These improvements were achieved through the reduction of physician idle time. In a concurrent study made by a team of sociologists who did a before and after set of interviews with the physicians, it was concluded that physician morale increased because of the work done.

The increase in throughput that was possible, together with the increased time spent with patients, and the fewer physician hours actually scheduled for patient contact, meant that the systems analysis was responsible for providing the students with services that would have required approximately 2.2 additional physicians operating under the old system. This meant a saving in excess of $50,000 in the first year in physicians' salaries alone, and if one also includes the support services that these two physicians would have required, this figure for savings would increase substantially.

The complete systems analysis used a simulation model to examine the effect of various strategies for scheduling the appointments of patients and for examining the effect of different working schedules for the physicians. Runs were compared on the basis of patient waiting time and physician idle time, the two most sensitive measures of effectiveness. Based on these criteria, the medical staff reviewed the results and decided on a scheduling pattern for themselves and their patients for the following academic year.

The key to the success of the enterprise was twofold: first, the entire system was analyzed by a team including sociologists, physicians and administrators as well as engineers; and second, the analysts were cast in the role of supplying staff support to both the clinic administrators and the medical staff. The clinic administrators formulated alternative scheduling patterns based on the questions that the medical staff raised, then the simulation analysts ran the model to replicate this situation. The results of the various simulation runs were examined by the medical staff and the clinic management under the guidance of the simulation analysts. The resulting decision represented the needs of the medical staff and the preferences of the clinic administration, and it took advantage of the technological expertise of the simulation analysts. The details of how the model was developed, the manner in which data were taken and the kinds of results that were obtained from the model are presented in the remainder of this article, but it must be remembered that this simulation model was only one facet of the enterprise.

Description of Facilities

In the fall of 1970, the University of Massachusetts Health Service delivered primary health care to approximately 19,000 students on a compulsory prepaid basis. There were, in addition to the outpatient department that is of interest here, approximately 70 inpatient beds, a laboratory and x-ray facilities, an emergency room, a pharmacy, and a mental health clinic that is separately housed. The University Health Service also operates a health education program and an environmental health and safety program.

The outpatient department usually treats between 400 and [illegible]00 patients per day. About half of these patients see a physician on either an appointment or a walk-in basis. The remaining patients visit clinics such as the nurse-practitioner clinic, where four nurses deliver primary care under the direct supervision of a physician, or special purpose clinics operated by nurses for things such as immunizations, TB tests, allergies, warts, obesity, etc. During the fall semester of 1970, the Health Service had twelve full-time physicians on its medical staff. Because of duties relating to administration, the inpatient area, the nurse-practitioner clinic, "on-call" periods during the evenings and weekends, and other tasks, only 260 physician hours per week were made available in the outpatient department during regular clinic hours. The rotating schedule meant that no more than seven physicians could be available at one time.

The outpatient department of the Health Service at the University of Massachusetts has many problems in common with other outpatient medical care delivery systems. The rapid growth experienced over the past several years has resulted in conditions common to most overcrowded health care facilities. The alleviation of the following conditions was identified as the immediate target of the study:

1. There was a long waiting time for patients.
2. The professional staff felt overworked and harassed.
3. There was much confusion and crowding in the waiting rooms at predictable times (on Monday, Tuesday, and Friday afternoon).
4. The physicians were still seeing patients as long as an hour past closing time.
5. During the day, physicians were sometimes idle because patients did not always keep appointments scheduled several weeks in advance.

6. The current building was (and is) overcrowded as it was designed for a student body of 10,000 and is currently serving a student body of over 19,000 students.

Procedure

Analysis of the targets of the work revealed that the basic problem was congestion in a complex queueing system. The procedure developed to solve this problem was based on the assumption that improved management of demand, through an expanded appointment system, better resource management, and more efficient physician scheduling, would make the system function more effectively.

The first step was to estimate the "demand" on the system. Specifically, the demand was defined as the number of physician visits per week that would occur during the regular clinic hours in the 1970 academic year. The estimated demand was divided into two components which were termed "controllable" and "uncontrollable". The contollable component of demand was defined as those patients who made (or could be induced to make) an advance appointment for their physician care. The uncontrollable component of demand, or "walk-in" patients, was defined as those patients who arrived without notice. This latter category would include both "emergencies" and those patients whose need for medical care possibly could be postponed, but was not.

It was then necessary to estimate the number of patients who could be induced to make an appointment for their physician visits by estimating the size of the controllable component of the demand. It would then be possible to distribute the various appointment periods throughout the week in such a way that they would "complement" the walk-in demand. By scheduling more appointment periods during the periods of low walk-in demand, the appointment patients would "smooth" the load of physicians and facilities. Naturally, this distribution of appointments would have to take into account the pattern of arriving walk-in patients, which was known to be different on the various days of the week and which also changed hour by hour during the day.

The attempt to smooth the demand for physicians' services during regular clinic hours proceeded in two steps. The first step was to attempt to smooth the demand by day of the week. It was to be judged successful if there were a uniform number of patients arriving each day of the week. The second step was to smooth the demand across the hours of the day. The criteria of success of this step were the measures of effectiveness of the whole design procedure, the patient waiting time and physician utilization.

These two steps were performed separately. The first step, smoothing the demand over the days of the week, was performed by straightforward analysis of historical trends to produce estimates for the future. The second step of smoothing the demand over the hours of the day required a sophisticated Monte Carlo simulation model.

Alhtough this procedure has intuitive appeal it must still be recognized as a piecemeal attack that omits consideration of all ancillary services (except as their effectiveness may be enhanced by a "smoothed" demand). Considering the present state of the art, this piecemeal approach is the best that can be managed for this type of problem.

Development of the Simulation Model

The present model, schematized in Figure 1 simulates the operation of both the Physician and Nurse Practitioner Clinic.

The model was developed to date in five distinct stages. The first stage was the limited scope model named "ASIS" that simulated a single stage parallel queue (for a variable number of channels) to describe the Physician Clinic. This model had two levels of patient priority. The second model was an extension to a three level priority system for patients with the capability of experimenting with different priority rules and scheduling tactics. The third model was built to represent the Nurse Practitioner Clinic as a two stage, two priority, parallel queue system with a variable number of channels. The fourth model combined the second and third models into one where patients were allowed to cross over from the Nurse Clinic to the Physician Clinic. The fifth and present model is an extension of the fourth. In this model, the servers are allowed to switch functions at predetermined times to limit their patient load to any predetermined class (or classes) of patients. The program is streamlined with the use of modular subroutines to enable ease of modification.

The language used for modeling was a local version of GASP II (4). Since at the outset of the project the University of Massachusetts Research Computer Library contained no debugged and documented simulation language, GASP II was chosen for two reasons: (1) the author's familiarity with Fortran and his understanding of GASP principles; and (2) ease of implemention for the available CDC 3600 and simple modification and expansion in the future.

Since then, several additions and modifications have been made to the GASP II simulation package to satisfy our needs, and this extended version has been found very useful for other projects.

The GASP II (discrete, next event) simulation language contains two groups of Fortran subroutines. One group takes care of the filing and retrieving of simulation events while the other group serves the statistical functions of the simulation, i.e., sampling of distributions, collecting of statistics, etc.

The model builder has only to construct Fortran subroutines using GASP II conventions to model the system, and the execution of the

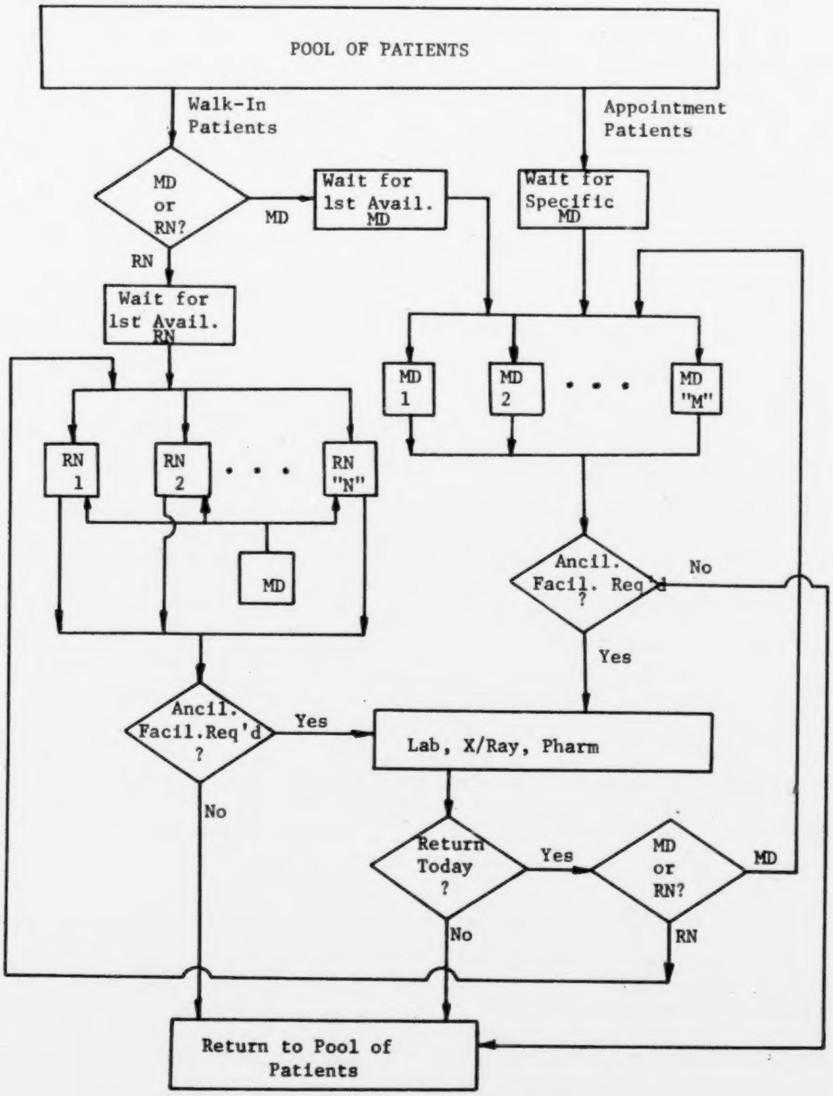

POOL OF PATIENTS
Walk-In Patients
Appointment Patients
MD or RN?
MD
Wait for 1st Avail. MD
Wait for Specific MD
RN
Wait for 1st Avail. RN
MD 1
MD 2
MD "M"
RN 1
RN 2
RN "N"
MD
Ancil. Facil. Req'd ?
No
Yes
Ancil. Facil.Req'd ?
Yes
Lab, X/Ray, Pharm
No
Return Today ?
Yes
MD or RN?
MD
No
RN
Return to Pool of Patients

Figure 1. Schematic Patient Flow Diagram Indicating Some of the Logic of the Simulation Program

simulation is then taken over by the GASP II executive package.

Data

Because the system was conceptualized as a complex queueing system, information was needed on the arrival patterns of patients and the way they spent their time in the system. This latter information was broken down into waiting time, the routing of patients through the system, and the amount of time required to serve their needs at each of the places in the system where they received service.

The data used to determine arrival patterns were taken from the encounter form that all patients fill out prior to any service they receive from the Student Health Service. After it is filled out by the arriving student, it is stamped with the date and time and placed with the medical record.

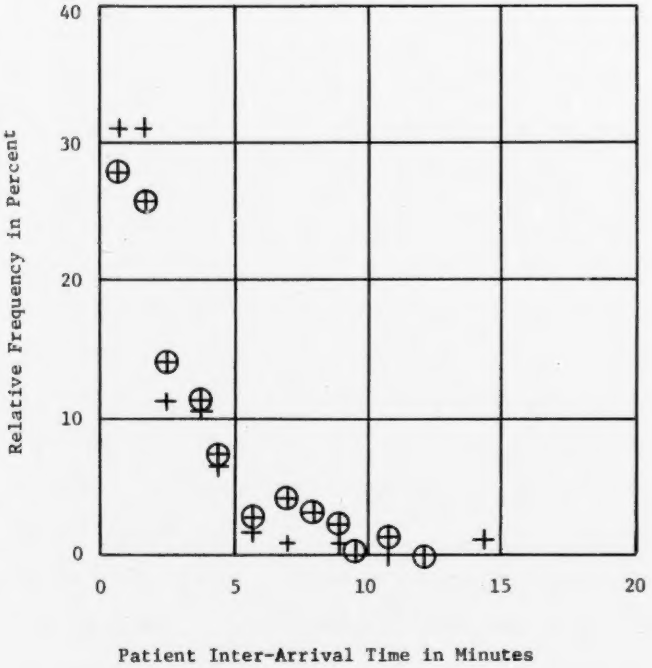
Data on the time physicians spent consulting with patients and the time patients spent in the laboratory, x-ray, etc. were taken during three separate weeks. Clerks were stationed near the entrance to physicians' examining rooms and other facilities and were furnished with date-time stamping clocks. Special record sheets provided for the purpose were stamped as each patient entered and left each service. These records were also time stamped and collected when the patient left the building. The information stamped on these forms gave an accurate account of the services the patient used and of the time necessary to provide the service in question as well as all waiting time involved. During two data-taking periods, the number of these special records collected agreed with the medical encounter forms within about 7 per cent; also, less than 5 per cent of the special forms provided to collect service time and routing data were unusable because of a missing arrival or departure stamp.

Simulation Runs

A word is in order concerning the interpretation of the results obtained from the simulation model. It quickly became clear that Monday, the day with the largest number of walk-in patients, and Thursday, the day with the largest number of appointment patients, were the days that were most sensitive to any scheduling tactics. Therefore, the bulk of the simulation studies were limited to situations found on Monday and Thursday.

The gross patient arrivals over the day (for one Monday and one Thursday) are shown in Figure 2. Figure 2 illustrates that the patient arrivals over the entire day are distributed in a negative exponential form. From this it was assumed that this form of distribution could be used throughout the day to generate walk-in arrivals even though the mean value of the distribution was changed hour by hour to correspond to the observed values.

The arrival rates for each hour of each day were available from the arrival date-time



+ Monday, April 6, 1970; $\bar{x}$ = 2.167, s=2.402, n = 237

+ Thursday, April 9, 1970; $\bar{x}$ = 2.626, s = 2.838, n = 202

Figure 2. Frequency Distribution of Patient Inter-Arrival Time in Minutes

stamp made on all medical encounter forms. This arrival pattern is illustrated in Figure 3 which shows the average number of all patients entering the Health Service to receive care for each hour of the day for Mondays and Thursdays during the fall semesters of 1969 and 1970. The similarity of the pattern between Monday and Thursday data demonstrates that there was little biasing effect of class hours, which tend to be scheduled at the same hours on Monday, Wednesday, and Friday, or at the same hour on Tuesday and Thursday. Although the 1970 data were not available when the analysis was performed, it is presented here to show that stability of the pattern.

The arrival pattern from 1969 was used to generate the walk-in patients for the Monte Carlo simulation model. Operationally, the arrival pattern was incorporated into the simulation model as inter-arrival times, and the parameters of this distribution were changed during each hour of simulated time. By this process, we were reasonably assured that the arrival pattern of the walk-in patients would replicate the pattern of walk-in patients which would actually occur.

The consultation times (service times) that physicians spent with patients were measured in three separate categories. Those categories were for appointment patients, walk-in patients, and the time required for "second service". These distributions are shown by the histograms in Figure 4A, 4B, and 4C.* The sample mean and sample standard deviation for appointment service times were found to be 12.74 minutes and 9.56 minutes and for walk-in service times were found to be 9.61 minutes and 7.48 minutes respectively. The values actually used in the Monte Carlo simulation were generated from a log-normal form and resulted in a distribution whose mean and standard deviation were 12.35 minutes and 9.05 minutes for appointment patients and 9.57 and 8.22 minutes for walk-in patients. These values correspond very closely to the values of 12.6 and 9.8 minutes respectively which were reported for two Air Force Ambulatory care facilities, and which were used by Fetter and Thompson in a portion of their study (5). The Nuffield study of ambulatory care facilities in England also reports similar figures (6); the average consultation time for new patients visiting physicians whose specialties roughly correspond to a practice of Internal Medicine is 11.8 minutes.

The examination of ten days of data showed that approximately fifteen per cent of the patients who see physicians are sent elsewhere in the clinic (e.g., laboratory, x-ray, etc.) and return to see the same physician again on the same day. A log-normal distribution with a mean and variance of 15.54 and 11.09 respectively was found to be an appropriate model for

---

* Service times in the medical practice are generally best described by either a Gamma or Lognormal distribution. Since generation of variates from the Gamme distribution is time consuming, the Lognormal form was used here.

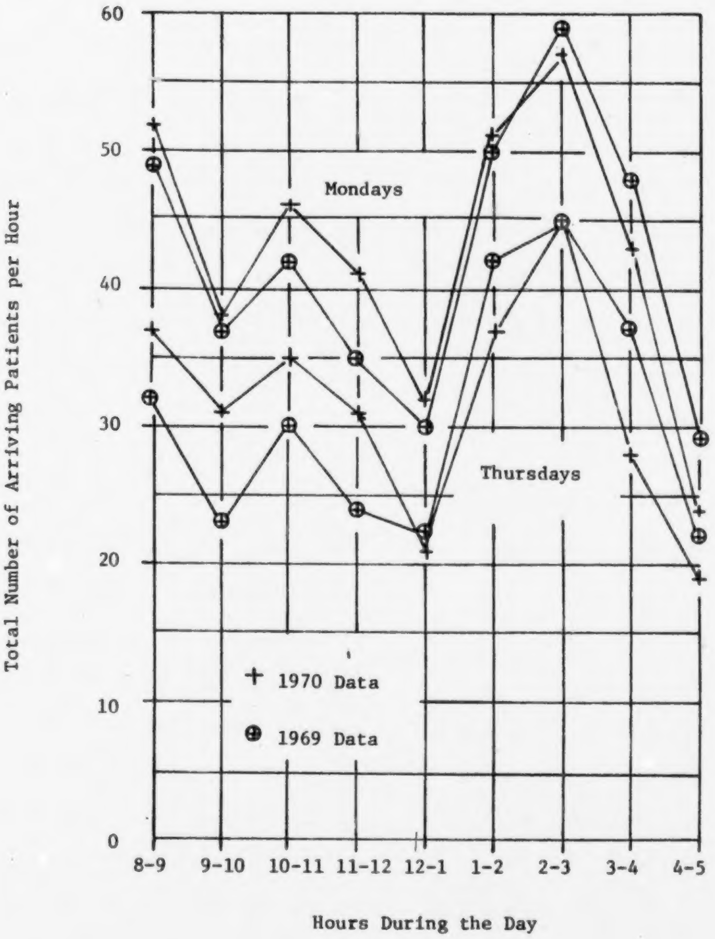


Figure 3. Hourly Arrivals at Student Health Service (Monday and Thursday Averages; Fall Semester 1969 and 1970)

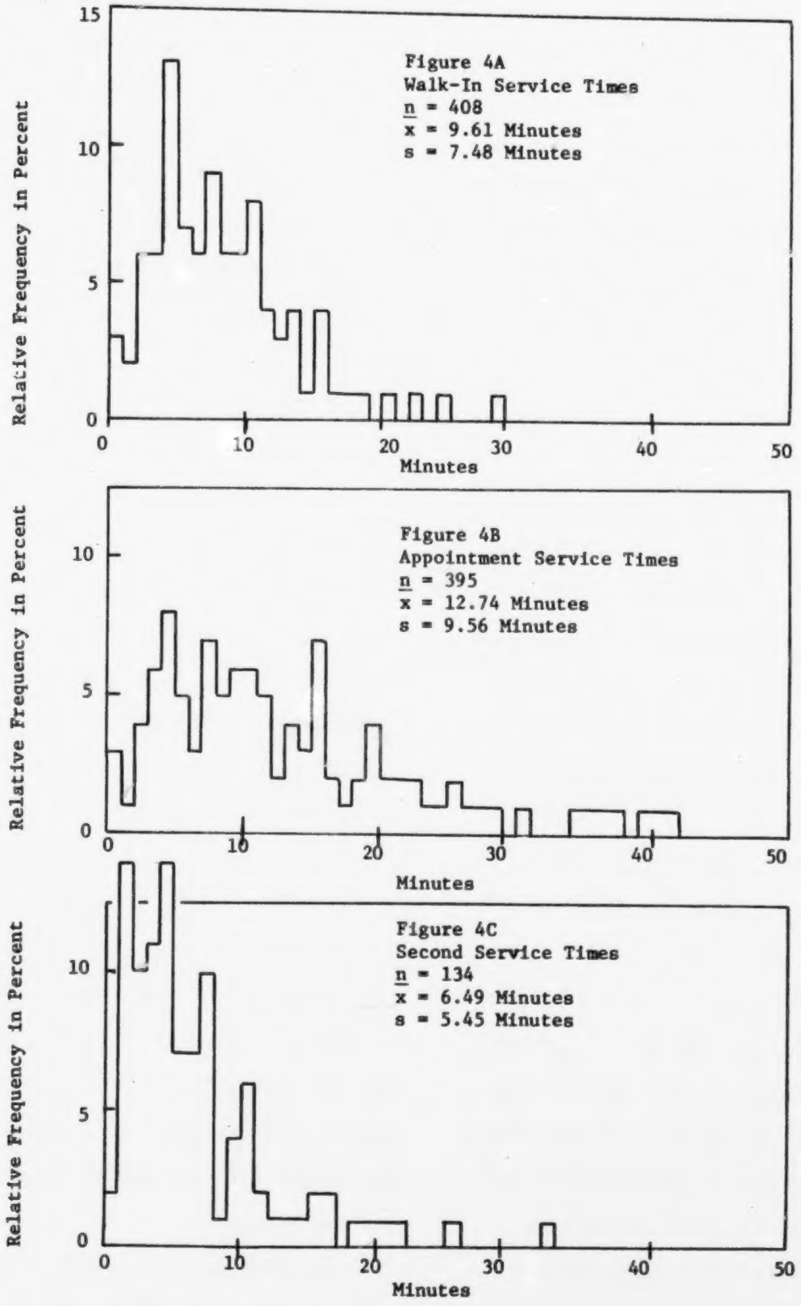


Figure 4. Histograms of Service Times

the elapsed period of time. These patients returning to see the physician were observed to interrupt the flow of new patients. A return visit to a physician seen earlier in the same day was termed "second service"; the sample mean and standard deviation were 6.49 minutes and 5.45 minutes. The values used in the Monte Carlo simulation were generated from a log-normal form and resulted in a distribution whose mean and standard deviation were 6.41 minutes and 4.91 minutes. No published data were found for comparison purposes.

In the actual operation of the clinic, the physicians see patients in a sequence governed by three priority considerations. First priority is given to emergency patients entering the system and patients who are returning from a visit to the laboratory, x-ray, etc. to see the same physician they have already seen earlier on the same day. Second priority is given to patients who have made an advance appointment with a specific physician. Last priority is given to walk-in patients who are then seen on a first-come, first-served basis. The walk-in patients are seen by any physician as soon as be becomes free of higher priority work. Most physicians use two examining rooms; a patient is being seen in one room while the next patient to see the physician is being prepared in the other. When a physician finishes with one patient, the priority system is used to select a patient for the examining room just vacated. This priority system is administered by a nurse who controls the flow of patients through release of medical records to individual physicians in the proper order.

Within the Monte Carlo simulation model, the priority rules are replicated by the use of two "files" for each physician and one "file" common to all physicians. There is a "priority file" and an "appointment patient file": for each physician, and the "walk-in patient file" is held in common. Each time a physician completes a service to a patient, the files are searched to locate a patient to fill the examining room just used, while the physician sees the patient already waiting in the second examining room. The files are searched in the following order: priority file, appointment file, and walk-in file.

During the operation of the simulation, information was collected on waiting time for appointment patients, walk-in patients, and all patient who undergo "second service". In addition, the simulation collected information on physician utilization and the amount of time beyond the end of regular clinic hours that was required to service all the patients who have arrived during regular clinic hours. These results were displayed in histograms with the mean value, standard deviation, maximum and minimum values.

The effects of two decisions were examined on the basis of the results obtained from the simulation. The first decision was the hours during the day it was best to schedule the physicians, and the second decision involved which of the hours scheduled were to be set aside for appointment patients. The selection of the best schedule of physician hours and the best time for appointment periods proceeded in three steps.

First, an intuitively attractive set of appointment periods was selected that approximately complemeneted the known hourly arrivals of walk-in patients. In the second step, this appointment pattern was held constant and the number of physicians was changed across various hours of the day within daily resource constraints (52 physician-hours, and a maximum of seven physicians at one time). The third step was to hold constant the best physician schedule found in step two, and then to go back and rearrange the appointment slots in an attempt to improve the solution. The second and third steps were repeated in an attempt to secure additional improvements, but none was obtained.

In general, the best physicians' schedule found was that seven physicians should work during the last six hours of the normal eight hour day. In actual practice, this pattern had minor deviations occasioned by the need to stagger the schedules of the physicians to accommodate lunch hours, coffee breaks, and a period for "rounds" in the inpatient area. The clinic was kept open nine hours per day to accommodate the daily eight hour working schedules of the physicians.

Table 1 presents the arrangement of appointment periods by hour of the day and by day of the week that produced the best simulation results when used with the above physician schedules.

The waiting times of appointment patients are relatively insensitive to how the appointment slots are arranged throughout the day because of the priority these patients are given. The waiting time of walk-in patients is highly sensitive to the arrangement of appointment slots through the day and therefore the waiting time of walk-in patients was the most useful criterion to use to make the decisions. It was also found that the number of minutes the clinic runs overtime is sensitive to the pattern of arrivals.

In general, it was found that provided a queue is built up early in the day by either scheduling fewer physicians at the beginning of clinic hours or by scheduling a group of early appointments, the physicians' idle time is relatively insensitive to the way the appointment periods are arranged throughout the day. Proof of this finding was first reported by Welch and Bailey (7) in 1952.

Comparisons between Predictions and Performance

A week or two after the best simulation results were implemented, minor adjustments of some physicians' schedules and appointment patterns were made. After a two month period of operation, data were then taken on service

TABLE 1

Hourly Schedule of Appointment Periods
Available During The Week

Hours	Mon	Tues	Wed	Thurs	Fri	Hourly Totals
8-9	9	7	7	9	7	39
9-10	21	21	21	21	21	105
10-11	13	14	14	15	14	70
11-12	7	7	12	13	12	51
12-1	0	0	0	0	0	0
1-2	10	15	15	17	13	70
2-3	12	12	17	18	18	77
3-4	17	17	13	17	17	87
4-5	7	7	7	6	7	34
Daily Total	96	100	112	116	109	533

times, patient routing, and patient waiting times. The routine management information system based on the encounter form operated constantly, and it yielded arrival data by hour of the day separately for each day.

The real test of the accuracy of the methodology lay in the comparison of predicted and actual outcome measures. However, since the predictions were based on service time data from the previous year, a follow-up study was made to determine if these data were stable from year to year. In this case, examination of service time distributions and patient routing provided this assurance; the follow-up data verified the base data. Furthermore, the evaluation of the control of patient arrivals over the days of the week has been given earlier and rests on the data presented for "smoothed" daily arrivals for the days of the week shown in Figure 5 and on the stability of the pattern of hourly arrivals between 1969 and 1970 shown earlier in Figure 3. Since this evidence was judged satisfactory, the way was cleared to examine the comparison of predicted and actual values of waiting time as a measure of success of the overall methodology.

The two main measures of the model's validity were the patient waiting times and the amount of time necessary for the clinic to remain open to serve the remaining patients after closing time. Initially, the model provided results which were much better than those of the real system. In checking our input data and assumptions, we discovered two discrepancies:

(1) the sum total of tasks measured did not add up to the length of the workday; and

(2) we discovered the system acting somewhat differently when the data takers were visible as compared to a normal day.

The first discrepancy was attributed to the fact that as in all labor-intensive work, allowances have to be made for fatigue and factors beyond operator control and the second was attributed to the classical "Hawthorne" effect which means that data collected with the subject's knowledge tends to show better performance than is the case under normal circumstances.

The allowance made to correct the above was to simulate equally lengthened coffee and lunch breaks for the staff. This allowance produced simulated results where waiting time is compared for five sets of values. The sets of simulation results shown are: first, the case where all physicians arrive at the clinic on time and leave and return promptly from coffee breaks and lunch. In thic case, there is no time "lost" from treating patients. The second case is where all physicians lost twenty minutes per day from scheduled clinic duties (an aggregate loss of 140 physician minutes per day); the third case is similar to the second case except the physicians lose 40 minutes per day each from scheduled clinic

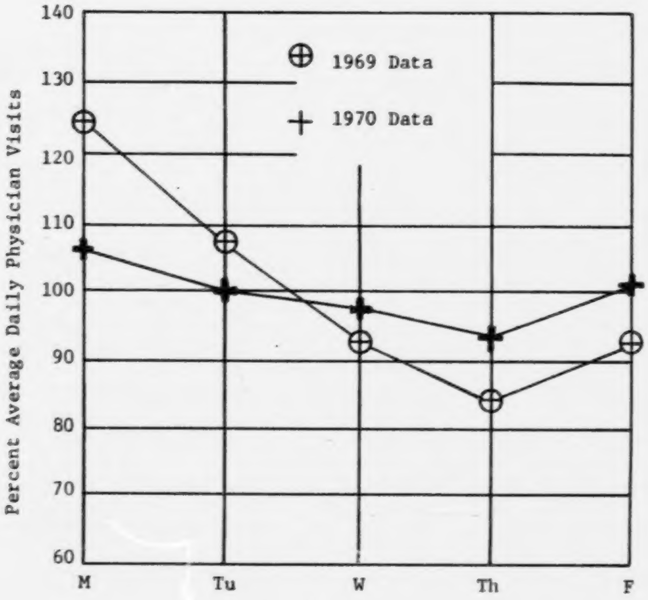


Figure 5. Effect of "smoothing" physician Visits by Day of the Week

duties (an aggregate loss of 280 minutes per day). The fourth and fifth sets of results further increase the loss of physician time to 60 and 80 minutes per physician, which aggregated to 420 and 560 minutes respectively.

In Figure 6, the data from Table 2 have been ploted for Monday and Thursday. This figure shows how the waiting time for appointment patients, walk-in patients and second service changes with the amount of physician "lost" time when the patient load on the clinic remains constant. The data for Monday show that walk-in waiting time is larger than appointment waiting time, and both these values increase for increased values of "lost" time. The Thursday data, the day with the greatest number of appointments, show the same general trends except that walk-in waiting time is less than appointment waiting time for low values of "lost" physician time. This phenomenon appears when the system is not congested and is the effect of physicians using two examining rooms. Use of two examing rooms makes an appointment patient's priority one of "second" in line for a particular physician while a walk-in is taken by any physician as soon as he is free.

Figure 6 shows that the parameters of the distributions of waiting times produced by the model are a function of the physician time that is "lost" to the clinic. The simulated results shown for "no lost time" are an idealized situation that can be used to set lower bounds on waiting times. The simulated values produced for the various amounts of waiting time provide estimates of what is likely to happen to waiting time under these various conditions of "lost" time. It should be pointed out that a "corridor consultation" between two physicians, or an emergency phone call would result in "lost" time to the clinic and therefore would have the same effect on the results obtained from the model as would tardiness and extended coffee breaks.

The entire distribution of simulated values whose mean values agreed best with the actual mean values are shown on the same graph in Figure 7 A, B, and C. Examination of this figure reveals that the simulation model produces distributions of waiting times very close to the actual values. Of particular interest is the fact that all three sets of predicted and actual distributions conform closely across their entire range of values.

The agreement between the form of the predicted and actual values shown in Figure 7, A, B, and C leads to the conclusion that the model behaves in much the same manner as the real world. Efforts are continuing to refine the model to obtain even better predictions and to give deeper insights into the operation of the real system.

It is felt that the next step is to examine the "state sensitivity" of arrival rates

TABLE 2

Simulated Waiting Times for Patients For Various Amounts of "Lost" Physician Time

		Amount of Lost Time Per Day Per Physician				
		0	20	40	60	80
Walk-In Patients	Mon	13	18	28	34	60
	Thurs	12	12	22	30	51
Appointment Patients	Mon	13	17	20	25	29
	Thurs	12	14	21	26	32
Second Service Patients	Mon	10	10	13	17	20
	Thurs	10	10	15	16	19
Aggregate Lost Time (Min./Day)		0	140	280	420	560

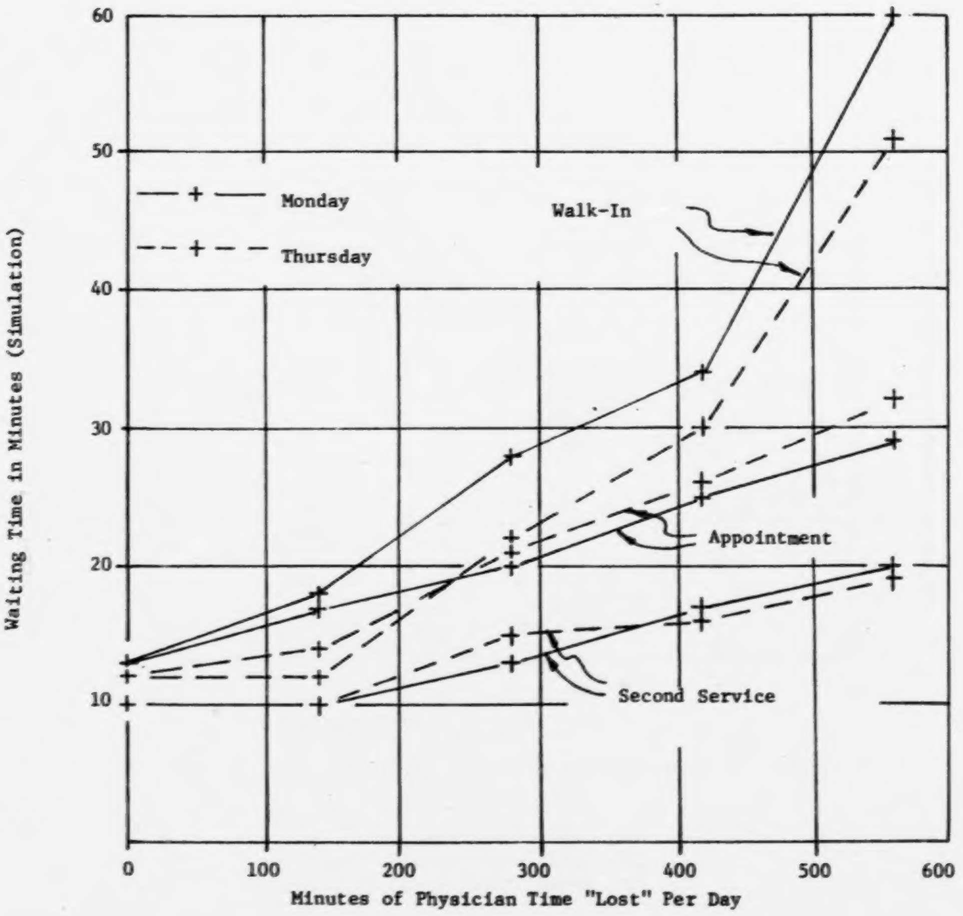


Figure 6. Simulated Waiting Time as a Function of the number of Minutes of Physician Time "Lost" Per Day

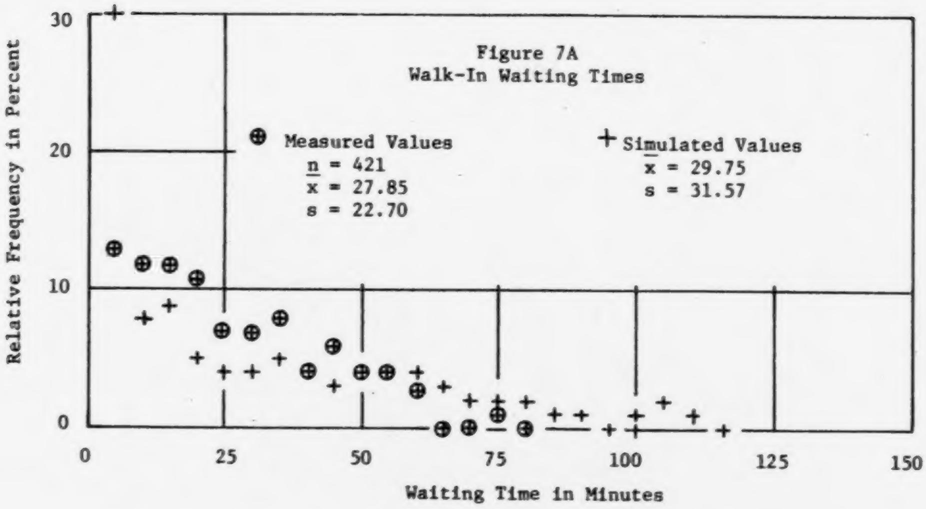


Figure 7. Comparison of Simulated and Actual Waiting Times

7A. Walk-In Waiting Times
7B. Appointment Waiting Times
7C. Second Service Waiting Times

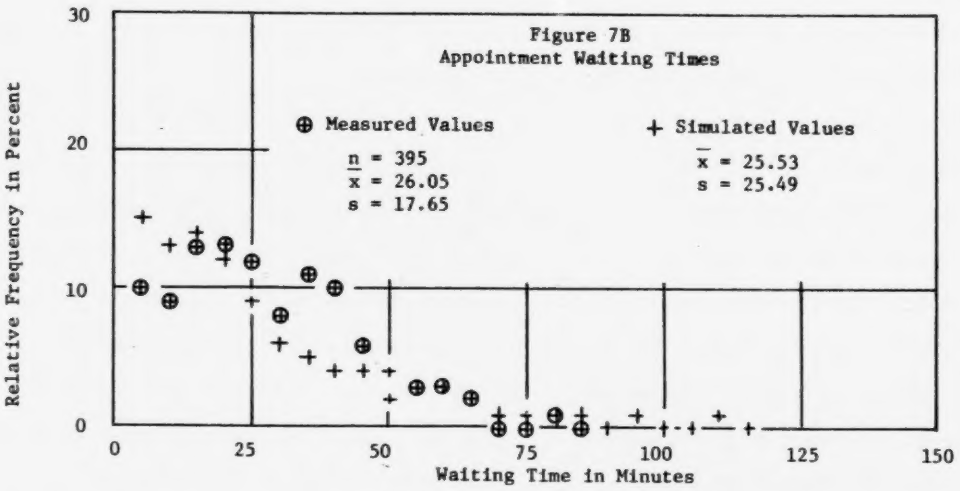


Figure 7B
Appointment Waiting Times

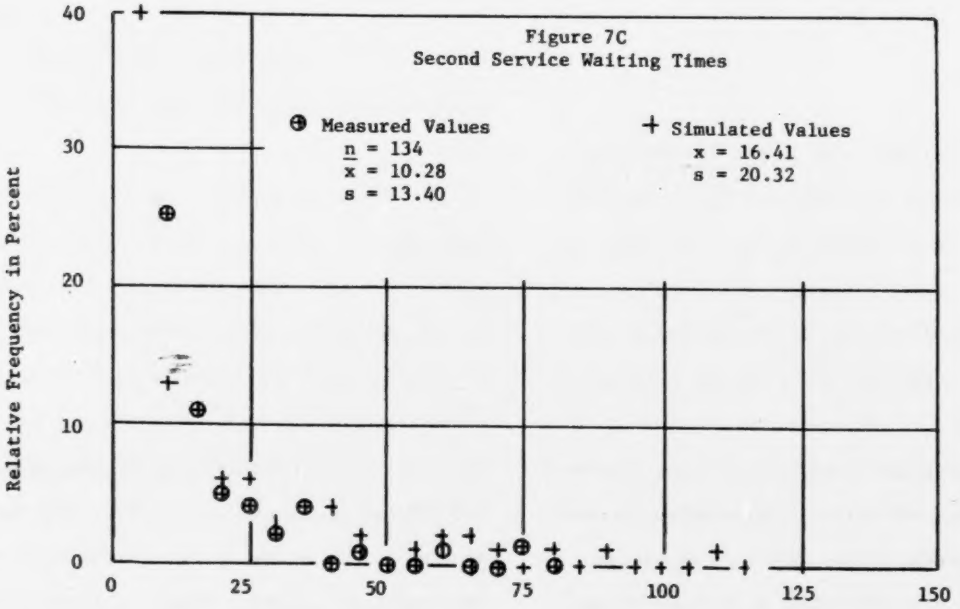


Figure 7C
Second Service Waiting Times

and service times. It has been observed that when the system is congested some of the arriving patients tend to look around the waiting room and then depart without entering the system. It has also been observed that in the last hour of the day, or when the physicians perceive the waiting room is crowded, they tend to reduce the length of their service time. Both of these cases illustrate that the current "state" of the system appears to have an effect on system parameters. It will be of interest to determine simple and straightforward techniques to measure these effects, apply them in the model, and then determine their effect on the predictive capability of the model.

Results

The data generated by the simulation model were thoroughly analyzed, and it was decided to implement the system changes that were studied. After the changes were accepted and considered routine by the operating staff, data was collected on the actual operation of the system. It was found that the number of patients that were seen by physicians increased by 13.4 per cent with a corresponding decrease of 5.1 per cent in the number of physician hours that were allocated to the walk-in an appointment clinics. In addition, interviews with the physicians confirmed that with the new operating policies, less overtime was required to finish treating those patients who remained in the system after clinic hours were closed. Data was not available to document the physicians' comments due to "student disorders" that were common during the end of the school year.

Another effect of the changes that were implemented was an increase of 5.0 per cent in the overall average time that patients spent with physicians. This increase was due to the increase in the number of patients that were seen by appointment. For both 1969 and 1970, the mean consultation time was 12.7 minutes and 9.6 minutes respectively for appointment and walk-in patients. The increase in throughput of 13.4 per cent and the average increase in service time of 5 per cent were based on a staff level of twelve full-time physicians; and this makes these increases of substantial value.

If these increases had had to be provided under the old system, it would require approximately 2.2 additional physicians. The fact is that the increased service was provided by the same staff and actually used by the student population which increases in size each year. At an average wage of approximately $25,000 for each physician in 1970, this results in a saving in excess of $50,000 the first year in physicians' salaries alone. Substantially more savings can be attributed to this analysis if the salaries of support personnel and equipment charges, hiring costs, fringe benefits, etc. are included.

The waiting time for both walk-in and appointment patients changed from 1969-70 to 1970-71, but the changes occurred in such a way that the overall average waiting time remained the same. The mean waiting time for walk-in patients decreased from approximately 38 minutes to 28 minutes, and the mean waiting time for appointment patients increased from approximately 12 minutes to 26 minutes. Taking into account the increase in the proportion of appointment patients in 1970-71, the weighted average is approximately 27 minutes, which is the same as the weighted average for 1969-70.

In a concurrent study performed by two sociologists, in which the physicians were interviewed both before and after the changes described took place, it was concluded that the physicians' morale improved.

Concluding Remarks

It is widely accepted that simulation modeling is as much an art as it is a science. The builder of a model must combine all the basic modeling elements in such a way that the finished product performs as much like a real world system as possible. In practice, this seldom happens in a direct and straightforward way. Ordinarily, a crude model is constructed which then goes through a series of refinements until the resulting model resembles certain aspects of the real world closely enough to be useful for decision making. Then the model is frequently used to play the "what if" game. The analyst uses the model to investigate what would happen if certain parameters in the real world were deliberately changed or happened to change. For example: How would the system respond if the demand doubled? What would happen if a physician were sick? And so on.

As more and more questions are investigated with the simulation model, adaptations to the basic model have to be made. The analyst finds himself in the position of not having a single simulation model but rather has an entire family of models, many of which may have been patched together in a hurry to be used only once and discarded; others are used over and over. When an analyst finds that one of these adaptations is used several times and more usage is forseen, it may become desirable to spend some time reprogramming the model to add new features and use the opportunity to improve the elegance of the programming.

We have conceptualized this sort of model development on a diagram in Figure 8. As one proceeds horizontally on this diagram, one sees the basic changes that are incorporated into the model to make it resemble reality more closely or comprehend a larger portion of the system. As one moves in the vertical direction on this diagram one may see growth or adaptations of a basic model type, each adaptation being identified with a specific question being asked. It is the authors' experience that a

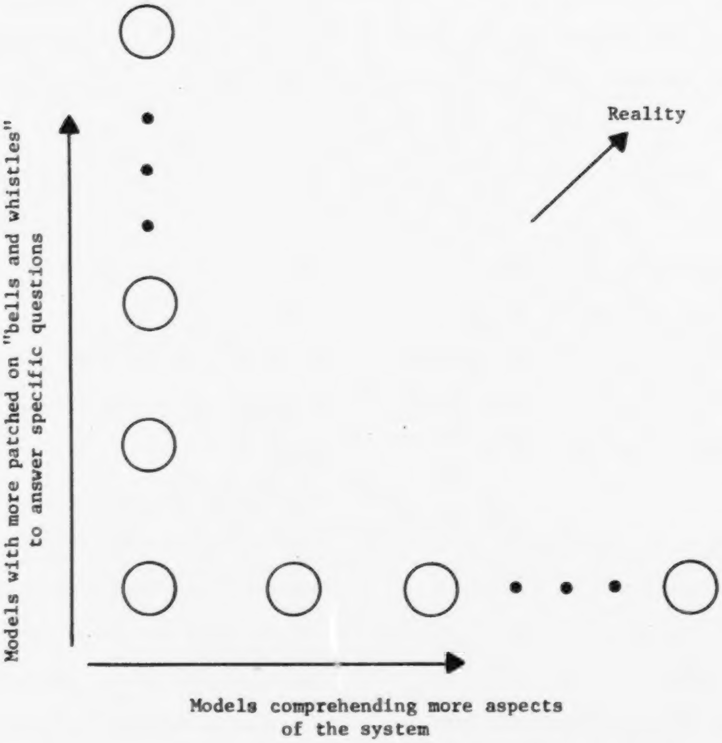

Reality
Models with more patched on "bells and whistles" to answer specific questions
Models comprehending more aspects of the system

Figure 8. Model Development

basic model type can be expanded in the vertical direction through added on features until it becomes so cumbersome that it is difficult to use efficiently. This may be reflected in such things as unnecessarily elaborate data requirements, extreme running times, and exceeding the capabilities of the available computer configurations. In cases like this, it becomes necessary to invest some time in a programming reorganization to achieve more efficient execution. In this case, a new version of the program would be written which is represented in Figure 8 by a step horizontally.

In practice, when a new question is to be investigated, there is always a conflict between the deisre to accommodate this question with a quick modification or to make a basic change in the model. The quick addition of a new feature to the current version of a program secures the answer to one more new question. A basic change in the model allows this new feature to be included in the model in a way that improves the efficiency as well as the flexibility of the program. It is unnecessary to say that the quick modification route is the one we have usually taken.

The point to be made from this examination of the philosophy of modeling described in Figure 8 deals with the problem of model validation. This diagram shows that the model of any real world situation has many versions, and it is neither economic nor is it possible to validate all the versions by comparing model output to real world data.

Suppose, for example, as in our present case, a comparison of one version of our model with real world data produces results as shown in Figures 7, A, B, and C. What then is the status of the validation of various adaptations of that model? Our view agrees with Naylor (8) that the validation of simulation models is mostly a matter of confidence and faith - not just a matter of mathematics. We believe that mathematics plays its part; it is necessary to compare some basic form of the model with the real world in a careful and systematic way using whatever mathematical tools may be appropriate and available. Once this is done, we feel it is not very fruitful to continue to go over this ground each time the model is adapted to a new problem. Once a model has been validated with real world data, we feel that the intuition of the real world system managers is adequate validation for adaptations.

We feel that the results of this study indicate the methodology presented for demand smoothing and the scheduling of physicians and their appointment patients is successful in this application. The gains in efficiency that we documented were substantial, and they were all in the right direction. We believe additional work will produce further gains. This success, we believe, was due in large part to the predictive capability of the simulation model. The model was "tailor made" to fit this facility.

A generalization emerges from this work when it is considered together with a study of the literature and observation of other outpatient facilities. It is that the methodology described here could be used to good effect on other outpatient facilities. The main obstacle to the widespread use of this approach is the cost and effort necessary to construct realistic simulation models. We believe that since all the published models of outpatient clinics follow the same general form, that is, queueing models solved by Monte Carlo simulation, it would be possible to develop a generalized model that is sufficiently flexible to overcome this obstacle. We are working on this task.

REFERENCES

1. Averill, B.W., Rising, E.J., McBride, T.C., Gage, R. W., and Piedmont, E.B. "The Outpatient Care Delivery System - A New Approach. I. Developing a New System." Journal of the American College Health Association 20: (334-339), June 1972.

2. Rising, E.J., Averill, B.W., and Baron R. "The Outpatient Care Delivery System - A New Approach. II. The Effects of Operations Research." Journal of the American College Health Association 20: (June 1972), 339-344.

3. Baron, R., Rising, E.J., Averill, B.W., and Los, T.J. "The Outpatient Care Delivery System - A New Approach. III. A Computerized Appointment System." Journal of the American College Health Association 20: (June, 1972), 344-350.

4. Pritsker, A.A.B., and Kiviat, P.J. Simulation with GASP II. Prentice-Hall, Inc., 1969.

5. Fetter, R.B. and Thompson, J.D. "Patient Waiting Time and Doctors' Idle Time in the Outpatient Setting." Journal of Health Services Research, Summer 1966.

6. Nuffield Report. Waiting in Hospital Outpatient Departments. Oxford University Press, 1965.

7. Welch, J.D., and Bailey, N.T.J. "Appointment Systems in Hospital Outpatient Departments." Lancet, 1952.

8. Naylor, T.H., and Finger, J.M. "Verification of Computer Simulation Models." Management Science, Vol. 14, No. 2, 1967.

Session 5: Simulation Methodology II
Chairman: Michael, Stonebraker, University of California

This session focuses on new techniques to assist practitioners of simulation in obtaining desired results efficiently. Many such users are attempting to find optimum performance of a simulated system. In this situation, the problem of selecting a procedure to search for the best choice is a challenging one. Two papers in this session compare alternate strategies for attacking this question. Other users face the task of finding confidence intervals for quantities obtained from simulation experiments. This job is often complicated by statistical dependence of successive observations. The third paper in this session suggests a way around this difficulty by utilizing properties found in many stable stochastic systems.

Papers
"Constrained Sequential-Block Search in Simulation Experimentation"
William E. Biles, University of Notre Dame

"Optimization of Simulation Experiments"
J. W. Schmidt, R. E. Taylor and V. Chachra,
Virginia Polytechnic Institute and State University

"A New Approach to Simulating Stable Stochastic Systems"
Michael A. Crane, Donald L. Inglehart,
Control Systems Corporation and Stanford University

Discussants
Grace Carter, RAND Corporation
Averill Law, University of California

CONSTRAINED SEQUENTIAL-BLOCK SEARCH IN SIMULATION EXPERIMENTATION

William E. Biles
Department of Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Abstract

This paper describes the application of sequential-block search techniques to simulation experimentation with constrained systems. Two basically different approaches are examined. One approach combines designed experiments, multiple regression, and mathematical optimization to predict a constrained optimum solution, which is then checked by further experimentation in the region of the predicted solution. A second approach employs a sequential optimum-seeking technique, such as gradient search or sequential simplex search, modified to accommodate constraints. These techniques are illustrated with a simple inventory system modeled with the GASP-II simulation language. A comparison of the effectiveness of these approaches is presented.

INTRODUCTION

The objective of simulation experimentation is to determine the optimum response y^* of some function of unknown form

$$y = F(X), \qquad (1)$$

where y is some measure of system effectiveness and X is an n-dimensional vector of input variables, x_i, $i = 1, \ldots, n$. Simulation experimentation consists of controlling the levels of the input variables X at several distinct sets of values, observing the simulated response y at each X, and eventually selecting X^* so as to yield the most beneficial response y^*.

Most realistic systems require consideration of several system responses, y_j, $j = 0, 1, \ldots, m$. The most expedient approach to multiple-response simulation experimentation is that of constrained optimization. In this approach, one response, y_o, is designated a primary or objective response. The remaining responses y_j, $j = 1, \ldots, m$ become restrictions or constraints by placing specifications on their performance. The mathematical statement of this problem is as follows:

Maximize (or minimize) $y_o = F(X)$ (2)

subject

$$a_i \le x_i \le c_i, \qquad i = 1, \ldots, n \tag{3}$$

$$y_j = G_j(X) \le (\text{or} \ge)\ d_j, \quad j = 1, \ldots, m \tag{4}$$

where

X = n-dimensional vector of input variables, x_i, $i = 1, \ldots, n$;

x_i = value of the ith input variable;

a_i = lower bound on the ith input variable;

c_i = upper bound on the ith input variable;

F = objective function, of unknown form;

y_o = objective response variable;

y_j = jth response variable;

G_j = jth constraint function, often of unknown form;

d_j = specification on the performance of the jth system response y_j;

n = number of input variables in the simulation model;

m = number of secondary system responses.

Although much has been done to develop improved techniques for simulation experimentation, scant attention has been given to the constrained optimization problem. This paper examines two basically different approaches to simulation experimentation with constrained systems. One approach combines designed experiments, regression, and mathematical programming in a procedure for predicting a constrained optimal solution. This paper compares central composite and simplex lattice designs for their effectiveness in predicting an optimal solution. A second approach utilizes search techniques in seeking a constrained optimal solution. This paper compares gradient search with two direct methods, sectional search (one-at-a-time method) and accelerated sequential simplex search.

EXAMPLE PROBLEM

The problem used to compare these various techniques is a simple (R, r, T) inventory system. In this problem, a retail outlet sells a particular item for $65. The wholesale cost of this item is $40. There is an inventory carrying charge of $0.20 per dollar-year. If a customer demands a unit when it is not in stock, he will purchase it at a competing retail outlet. The outlet under study assigns a loss of $20 to each such lost sale. The inventory position (units in stock plus those on order) is reviewed every T time periods. If inventory position P is less than or equal to the reorder point r, an order is placed for R-P units. The cost of each review is $2 and the cost of placing an order is $3. The demand for the item is Poisson-distributed with a mean of five units per week. The procurement lead time is Erlang-distributed according to the relation

$$f_x(x) = \begin{cases} \dfrac{\mu}{(k-1)!}(\mu x)^{k-1}\, e^{-\mu x} & x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

with μ equal to 2 and k equal to 6.

The retail outlet wishes to maximize profits from retailing this item, but it must operate within the following conditions:

1. The stock-on-hand cannot exceed 60 units due to space limitations;
2. Only one review can be performed on any given day, and a review is required by management policy at least once every three months;
3. The manager wishes to have the average weekly lost sales not exceed 0.2 units.

This leads to the following constrained optimization problem:

$$\text{Maximize } y_0 = \$25\, y_1 - \$20\, y_2 - \$0.15344\, y_3 - \$3\, y_4 - \$2/x_3 \qquad (6)$$

subject to

$$\begin{aligned} 0 \le x_1 &\le 60 \\ 0 \le x_2 &\le 60 \\ x_1 &\ge x_2 \qquad (7) \\ 0.2 \le x_3 &\le 13.0 \\ y_2 &\le 0.2 \end{aligned}$$

where

y_0 = average weekly profit, \$;

y_1 = average weekly sales, units;

y_2 = average weekly lost sales, units;

y_3 = average stock-on-hand per week; units;

y_4 = average weekly orders;

x_1 = inventory position, R, units;

x_2 = reorder point, r, units;

x_3 = review period, T, weeks

This problem assumes a five-day week. Note the discrete nature of the independent variables. If x_3 is considered on a daily basis, all three independent variables x_i, $i = 1, 2, 3$ are discrete. Note also that the objective response function is expressed in terms of four response variables. Hence, there are five response variables which must be observed experimentally.

The simulation model for this problem is written in FORTRAN using the GASP-II simulation language [13]. The simulator used in this study consists of a MAIN program, an EVNTS subroutine, and four events subroutines DMAND, PEREV, RECPT, and ENDSM. These components provide the following functions:

1. MAIN
   - a. Initializes model variables.
   - b. Turns control over to GASP executive.
2. Subroutine EVNTS
   - a. Transfers control to the appropriate event subroutine.
3. Subroutine DMAND
   - a. Creates next demand in accordance with the Poisson-distributed arrival rate.
   - b. Tests stock level. The variable SALES is incremented by one if STOCK > 0 and SLOST is incremented by one if STOCK = 0.
   - c. Collects statistics on STOCK if a sale is made.
4. Subroutine PEREV
   - a. Checks inventory position P against reorder point r. If P ≤ r, the receipt of R-P units is scheduled in accordance with the Erlang-distributed procurement lead time.
   - b. Increments number of orders ORD by one if an order is placed.
   - c. Restores inventory position P to level R if an order is placed.
5. Subroutine ENDSM
   - a. Terminates simulation.

b. Computes weekly averages for the following quantities:

   1.) Stock, y_3,

   2.) Orders placed, y_4,

   3.) Sales, y_1;

   4.) Lost sales, y_2;

   5.) Profit, y_o;

A six-year or 312-week period of operation is examined for all experiments in this study.

DESIGNED EXPERIMENTS

Considerable attention has been given to using designed experiments in simulation experimentation. Burdick and Naylor [4], Hunter and Naylor [7], Mihram [9], and Schmidt and Taylor [14] provide excellent treatments of this subject. Most of these works suggest the use of a sequence of first-order experiments in moving toward an optimum, switching to a second-order design in the vicinity of the optimum. Montgomery and Evans [10] have evaluated several second-order designs for experimenting with simulation models.

This paper examines the use of two second-order designs for simulation experimentation, (1) a central composite design by Box [3] and (2) a simplex lattice design. The basic procedure used for this study is as follows:

1. A designed experiment consisting of a predetermined set of design points is performed with the GASP-II simulation model of the (R, r, T) inventory system. Each of the responses y_k, k=0, 1, . . . , m is observed and recorded.
2. A multiple linear regression program is used to fit quadratic models of the form

$$y_k = b_o + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{ii} x^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} b_{ij} x_i x_j \qquad (8)$$

   for each of the m+1 responses [8].
3. The fitted equations are used to formulate a constrained optimization problem as expressed by equations (2) - (4), which is solved using a computerized constrained pattern search procedure [11] based on the Hooke and Jeeves search method [6].

Central Composite Design

The central composite design for a system of three independent variables is shown in coded form in Table 1. Table II gives the actual values of x_1, x_2, and x_3 for the present problem. Observe that the radial points in the design are not exactly equal to the α values specified by the central composite design, due to the discrete nature of x_i, i = 1, 2, 3. The values y_o and y_2 are also given in Table II. The center point is thrice replicated to provide an estimate of lack-of-fit error. The central composite design provides $(2^n + 2n + 1)$ points, compared to the $[(n+1)(n+2)/2]$ coefficients in the quadratic model given by (8). For larger problems, the number of points in the central composite design considerably exceeds the number required by the quadratic model.

Simplex Lattice Design

A design that is very economical for use with quadratic models is the {n, 2} simplex lattice design. Myers [12] describes the use of simplex designs for first-order experiments. Figure 1 shows two and three-dimensional first-order simplex designs. The {n, 2} simplex lattice design follows directly from the first-order simplex design by placing a point at the mid-point of each edge of the simplex, as illustrated in Figure 2. This provides exactly the [(n+1) (n+2)/2] design points needed for estimating the quadratic model. A center point can be placed at the centroid of this system and replicated to provide a test of error due to lack of fit. Table III gives the design points and responses for a simplex lattice design for the (R, r, T) inventory problem.

Comparison of the Two Designs

To provide a comparison of the two designs, the data from Tables II and III were employed in a "canned" multiple regression package to fit quadratic equations of the form given by (8). The resulting equations were then used in formulating the constrained optimization problem which was solved using the "canned" pattern search. The results of these studies were as follows:

	Central Composite	Simplex Lattice
X	(60, 45, 12)	(49, 37, 9)
Profit (predicted)	$124.25	$122.43
Profit (actual)	118.52	121.49
Lost Sales (predicted)	0.011	0.036
Lost Sales (actual)	0.035	0.0

Hence, the simplex lattice design performs slightly better than the central composite design in this problem. The main advantages of the simplex lattice design, however, are those which contribute to its relative economy:

1. It uses exactly the [(n+1) (n+2)/2] points needed to estimate the quadratic model.
2. It develops directly from a first-order design.
3. It contains smaller simplices (refer to Figgure 2) which can be used to form a simplex lattice design in a sub-space of the experimental region around a predicted solution simply by performing the experiments corresponding to the edge mid-points for the simplex sub-space.

Disadvantages of the simplex lattice design are as follows:

1. It does not possess optimal statistical properties, such as minimum bias and minimum variance. Furthermore, no attention has yet been given to describing the mathematical properties of the design.
2. The orientation of the simplex in the factor space is left to the judgement of the experimenter. (The vertices of the design given in Table III closely approximate an orthogonal first-order simplex design given by Myers [12]).

SEARCH METHODS

An alternative to employing designed experiments in simulation experimentation is to use a search technique. These fall into one of two basic categories, (1) gradient methods and (2) direct methods. They can be made completely automatic by having a "canned" program compute the succession of observations in the search, or they can be made adaptive by having the experimenter examine the results after each block of experiments and plan the next block. The latter approach is likely to make more efficient use of computer time and is the scheme developed in this paper. A gradient search procedure is compared with two direct search methods, sectional (one-at-a-time) search and accelerated sequential simplex search. Each of these methods has the feature that experimentation proceeds in a sequence of blocks, allowing the experimenter to exercise his judgement as experimentation progresses.

Gradient Search

Gradient search is initiated by placing a set of experiments around a base point X_o to estimate the gradient. For a system of n variables, n+1 experiments must be employed in estimating the gradient, as given by the following expression:

$$\begin{aligned} &X_o \\ &X_1 = X_o + \Delta x_1 \\ &\;\vdots \qquad \;\;\vdots \\ &X_n = X_o + \Delta x_n \end{aligned} \tag{9}$$

After observing the n+1 responses $y_o, y_1, \ldots, y_n$, the experimenter can compute the gradient direction as

$$m_j = B_j \Big/ \left[\sum_{i=1}^{n} B_i^2 \right]^{\frac{1}{2}} \qquad j = 1, \ldots, n \tag{10}$$

where

$$B_j = \frac{\Delta y_j}{\Delta x_j} \tag{11}$$

Δy_j is the change in the response y caused by the incremental change Δx_j, with all other variables held at the X_o level.

Having determined the gradient direction, the next block of experiments is performed at uniform intervals along this direction. For constrained systems, the bounds given by (3) will limit the step in the gradient direction. This combination of a gradient-determining block and a step-determining block is repeated until an acceptable solution is found. Beveridge and Schechter [1] give an excellent presentation of this topic.

Table IV presents the results of a gradient search approach to the example (R, r, T) inventory problem. The requirement for discrete values of x_i, i = 1, 2, 3 somewhat complicated the selection of experiments in the step-determining blocks, so that in effect only a "near-gradient" direction could be followed. Nevertheless, gradient search is seen to be adequately effective as a simulation search technique. The search was terminated after block 8, because the indicated gradient direction would have caused constraint violation.

In Table IV, blocks 1, 3, 4, 6, and 8 are gradient-determining blocks. Blocks 2, 5, and 7 are step-determining blocks. In block 2, the best point along the gradient direction was (60, 20, 30). The gradient from this point, however, as computed from the results from in block 3, would have violated the upper bound on the variable x_1. Therefore, the decision was made to evaluate the gradient from the next best point in block 2, (40, 20, 34), which produced the results in block 4. This episode points out one of the difficulties in sequential-block experimentation, that subjective judgements must often enter the experiment selection process.

Sectional Search

Perhaps the simplest direct search method is that in which only one variable at a time is changed. By keeping n - 1 of the n variables fixed at some level, the remaining variable can be altered over its range. This process is repeated until an optimal solution is found.

Table V gives the results of a sectional search applied to the example inventory problem. Four experiments are used in each block, except in block 2 where the fourth experiment would have duplicated an experiment from block 1. In block 2, X_2 was varied from 18 to 36, since it could not exceed X_1, which was maintained at 40. In block 3, X_1 was varied from 38 to 56, since it could not fall below the value of X_2 at 36. In block 4, X_3 was varied from 4 to 16, since higher values had been shown in block 1 to be less profitable. The search was halted in block 6, since none of the experiments in the block produced results superior to the solutions observed in blocks 4 and 5.

Accelerated Sequential Simplex Search

A technique that appears promising for simulation experimentation is the accelerated sequential simplex search method [2], which is based on the sequential simplex method of Spendley, Hext, and Himsworth [15]. Instead of moving along one point at a time, however, this new technique employs a simplex of n+1 points in each successive block. The direction of movement is that from the worst point in the simplex through the centroid of the n remaining points. If the same direction is maintained in successive blocks, the movement accelerates in accordance with the following relations:

$$X_k' = X_k + 2h(X_c - X_w), \quad k = 0, 1, \ldots n \qquad (12)$$

where

X_k' = kth vertex in the next simplex,

X_k = kth vertex in the current simplex,

h = no. successive blocks in which the same direction is maintained,

X_w = point yielding worst response y,

$$X_c = [\sum_{j \epsilon S} X_j]/n, \qquad (13)$$

where S is the set of all points in the simplex other than X_w.

Figure 3 shows the progress of the standard sequential simplex search technique for a simple two-dimensional problem. Figure 4 shows the progress of the accelerated method for the same problem. Table VI presents the results from employing accelerated sequential simplex search with the (R, r, T) inventory problem. The worst point in each simplex is noted with an asterisk. The search was halted after block 6 because the indicated direction of movement to a seventh block was toward a region that had already been examined in block 5. Moreover, the maximum profit in block 6 was less than 0.6 percent higher than that in block 5.

Comparison of Search Methods

Of the three search methods examined here, the accelerated sequential simplex procedure yielded the best solution to the example (R, r, T) inventory problem. With respect to search efficiency, the simplex and sectional search procedures each required six sequential blocks, compared to eight blocks for the gradient procedure. The results are conditioned, however, on the somewhat arbitrary criteria which were used to stop the search.

The initial experiments by each procedure produced solutions that violated the lost sales constraint; however, moves that gave improved values of the objective response y_0 also reduced the extent of lost sales constraint violation. This outcome is not surprising, considering the relatively high cost of a lost sale. This is not the most realistic situation one could encounter, however, and the example problem is defective in that regard.

To summarize the procedures to apply in the face of constraints, the foremost rule is to initiate the search in the interior of the feasible region. The three search methods could then operate in the following ways:

1. In gradient search, select as a point along the gradient direction that point which yields the maximum value of the objective response y_0 without violating a constraint. The gradient-determining block would then be performed to establish the best direction from that point.
2. In sectional search, consider only those experimental points in each block which do not violate constraints, selecting that point which maximizes the objective response.
3. In accelerated sequential simplex search:
   a. If, for a simplex derived by letting $h \geq 2$, constraint violation occurs, set $h = 1$ and compute the next simplex.
   b. If, for a simplex derived by letting $h=1$, constraint violation occurs, select a point other than the worst point as X_W. Recompute a new simplex with $h = 1$.
   c. If rules a and b fail to yield a solution satisfying all constraints, curtail the search and adopt the best observed point as a solution.

It should be stressed that none of these methods produce a globally optimal solution. They are effective, however, in producing a very worthwhile solution.

CONCLUSIONS

This paper has discussed the use of sequential-block search tecnniques in simulation experimentation with constrained systems. Two basically different procedures have been examined, each of which is effective in locating an acceptable constrained solution. None of the techniques examined here assure a globally optimal solution, however.

Of the two second-order experimental designs studied, the simplex lattice design offers both economy and search effectiveness in simulation experimentation. There is much to be learned about this design, however, and additional research in both its theoretical and practical aspects is necessary. The approach of performing a designed experiment, fitting first or second-order response models, and applying a mathematical programming procedure in seeking a constrained optimal solution is definitely worthwhile for simulation experimentation.

Gradient or direct search is another practical and effective approach to constrained systems simulation experimentation. This approach is especially useful for complex systems, where the experimenter desires to exercise his own judgement after each block of experimentation. A technique that appears to be very promising for sequential-block experimentation is accelerated sequential simplex search. This technique retains the advantages of the standard sequential simplex search technique, including an effective direction-determining mechanism, and adds the capability for acceleration in a direction that consistently proves favorable. There is a definite need, however, to evaluate this technique for problems of dimension greater than three. Additional evaluation of movements near binding constraints is also necessary.

REFERENCES

1. Beveridge, G. S., and R. S. Schechter, Optimization: Theory and Practice, McGraw-Hill, New York (1970).

2. Biles, W. E., "An Accelerated Sequential Simplex Search Technique," (in review) AIIE Transactions.

3. Box, G. E. P., and K. B. Wilson, "On the Experimental Attainment of Optimum Conditions," Journal of the Royal Statistical Association, Series B, 13, (1951).

4. Burdick, D. S., and Naylor, T. H., "Design of Computer Simulation Experiments for Industrial Systems," Communications of the ACM, 9, 5 (1966).

5. Draper, N. R., and H. Smith, Applied Regression Analysis, John Wiley, New York (1966).

6. Hooke, R., and T. A. Jeeves, "Direct Solution of Numerical and Statistical Problems," Journal of the American Association of Computing Machinery, 8 (1961).

7. Hunter, J. S., and T. H. Naylor, "Experimental Designs for Computer Simulation Experiments," Management Science, 16, 7 (1970).

8. "Multiple Linear Regression," IBM Scientific Subroutine Package, International Business Machines, New York.

9. Mihram, G. A., "An Efficient Procedure for Locating the Optimal Simular Response," Fourth Conference on the Applications of Simulation, New York (1970).

10. Montgomery, D. C., and D. M. Evans, "Second Order Response Surface Designs in Digital Simulation," 41st National ORSA Meeting, New Orleans (1972).

11. Moore, C. F., C. L. Smith, and P. W. Murrill, "Multidimensional Optimization Using Pattern Search," IBM Share Library, LSU PATE SDA 3559 (1969).

12. Myers, R. L., Response Surface Methodology, Allyn and Bacon, Boston, Mass. (1971).

13. Pritsker, A. A. B., and P. J. Kiviat, Simulation with GASP-II, Prentice-Hall, Englewood Cliffs, N. J. (1969).

14. Schmidt, J. W., and R. E. Taylor, Simulation and Analysis of Industrial Systems, Irwin, Homewood, Illinois (1970).

15. Spendley, W., G. R. Hext, and F. R. Himsworth, "Sequential Application of Simplex Designs in Optimization and Evolutionary Operations," Technometrics, 4 (1962).

Table I

Coded Design Points for the Central Composite Design

Design Point	x_1	x_2	x_3
1	-1	-1	-1
2	-1	-1	1
3	-1	1	-1
4	-1	1	1
5	1	-1	-1
6	1	-1	1
7	1	1	-1
8	1	1	1
9	$-\alpha$	0	0
10	α	0	0
11	0	$-\alpha$	0
12	0	α	0
13	0	0	$-\alpha$
14	0	0	α
15	0	0	0

Note: For $n = 3$, $\alpha = 1.216$

TABLE II

DESIGN POINTS FOR CENTRAL COMPOSITE DESIGN FOR EXAMPLE PROBLEM

Design Point	R x_1	r x_2	T x_3	Profit y_o	Lost Sales y_2	Seed
1	46	36	2	$117.27	0.0	5461
2	46	36	18	118.35	0.11	5461
3	46	44	2	116.81	0.0	5461
4	46	44	18	118.31	0.11	5461
5	54	36	2	117.54	0.02	5461
6	54	36	18	112.68	0.23	5461
7	54	44	2	119.78	0.0	5461
8	54	44	18	122.31	0.01	5461
9	45	40	10	121.90	0.01	5461
10	55	40	10	122.88	0.01	5461
11	50	35	10	121.80	0.04	5461
12	50	45	10	121.43	0.0	5461
13	50	40	1	114.86	0.0	5461
14	50	40	19	125.50	0.02	5461
15	50	40	10	119.88	0.0	5461
16	50	40	10	125.84	0.02	1971
17	50	40	10	119.53	0.01	8433

TABLE III

DESIGN POINTS FOR SIMPLEX LATTICE DESIGN FOR EXAMPLE PROBLEM

Design Point	R x_1	r x_2	T x_3	Profit y_o	Lost Sales y_2	Seed
1	50	46	2	$116.93	0.0	5461
2	40	36	18	110.58	0.30	5461
3	50	26	2	115.78	0.11	5461
4	60	36	18	117.98	0.11	5461
5	45	41	10	120.35	0.01	5461
6	50	36	2	118.76	0.0	5461
7	55	41	10	119.36	0.0	5461
8	45	31	10	117.82	0.13	5461
9	50	36	18	118.24	0.10	5461
10	55	31	10	118.27	0.05	5461
11	50	36	10	121.73	0.02	5461
12	50	36	10	125.42	0.06	1971
13	50	36	10	121.10	0.08	8433

TABLE IV

GRADIENT SEARCH APPLIED TO (R, r, T) INVENTORY PROBLEM

Block	R x_1	r x_2	T x_3	Profit y_o	Lost Sales y_2
1	30	20	36	$50. 37	1. 72
	33	20	36	59. 99	1. 51
	30	23	36	50. 37	1. 72
	30	20	39	48. 39	1. 72
2	35	20	35	65. 69	1. 37
	40	20	34	81. 67	0. 98
	45	20	33	61. 32	1. 47
	50	20	32	65. 75	1. 37
	55	20	31	76. 08	1. 12
	60	20	30	91. 01	0. 74
3	57	20	30	85. 17	0. 93
	60	23	30	92. 26	0. 75
	60	20	27	87. 88	0. 82
4	43	20	34	80. 32	1. 01
	40	23	34	83. 55	0. 92
	40	20	31	85. 43	0. 90
5	38	23	28	74. 40	1. 18
	36	26	22	91. 54	0. 77
	34	29	16	104. 74	0. 44
	32	32	10	111. 98	0. 25
6	35	32	10	116. 39	0. 15
	32	29	10	105.27	0. 40
	32	32	7	118. 18	0. 20
7	34	33	7	120. 94	0. 13
	36	34	4	122. 70	0. 01
	38	35	1	112. 94	0. 0
8	39	34	4	120. 82	0. 04
	36	31	4	119. 86	0. 06
	36	34	7	121. 62	0. 08

TABLE V

SECTIONAL SEARCH APPLIED TO (R, r, T) INVENTORY PROBLEM

Block		R x_1	r x_2	T x_3	Profit y_o	Lost Sales y_2
	*	40	30	10	$119.95	0.058
1		40	30	25	95.18	0.72
		40	30	40	69.40	1.30
		40	30	55	49.94	1.70
		40	18	10	85.74	0.92
2		40	24	10	102.90	0.48
	*	40	36	10	120.38	0.022
		38	36	10	119.26	0.067
3		44	36	10	120.67	0.0
	*	50	36	10	121.73	0.022
		56	36	10	119.71	0.039
		50	36	4	120.19	0.019
4		50	36	8	122.38	0.035
	*	50	36	12	122.87	0.058
		50	36	16	119.60	0.074
		50	34	12	117.28	0.12
5		50	38	12	121.94	0.006
	*	50	40	12	122.87	0.019
		50	42	12	121.73	0.010
		46	40	12	121.71	.003
6		48	40	12	122.81	0.016
		52	40	12	121.93	0.0
		54	40	12	118.93	0.055

Note: * denotes best value of X

TABLE VI

ACCELERATED SEQUENTIAL SIMPLEX SEARCH WITH (R, r, T) INVENTORY PROBLEM

Block		R x_1	r x_2	T x_3	Profit y_o	Lost Sales y_2
	*	30	20	35	$ 48.74	1.75
1		34	21	36	62.95	1.44
		31	24	36	53.77	1.65
		31	21	39	51.63	1.65
		34	24	39	60.50	1.45
2		38	25	40	67.25	1.32
		35	28	40	57.98	1.54
	*	35	25	43	48.92	1.79
		35	25	32	68.76	1.32
3		39	26	33	87.27	0.84
		36	29	33	79.14	1.02
	*	36	26	36	68.73	1.31
		38	28	19	106.31	0.46
4		42	29	20	105.46	0.44
		39	32	20	107.72	0.42
	*	39	29	23	104.12	0.46
	*	41	31	2	118.82	0.055
5		45	32	3	119.28	0.016
		42	35	3	120.27	0.0
		42	32	6	123.94	0.039
		45	35	6	124.64	0.045
6		49	36	7	122.49	0.0
		46	39	7	120.90	0.0
	*	46	36	10	120.34	0.003

Note: * denotes worst point, X_w

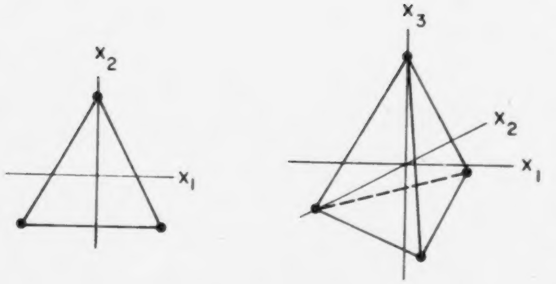
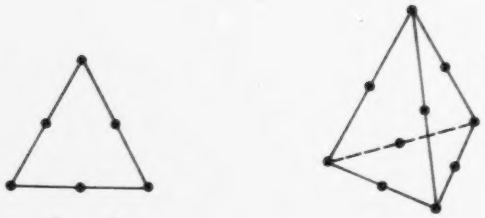


Figure 1
First-Order Simplex Designs



{2,2}
SIMPLEX LATTICE

{3,2}
SIMPLEX LATTICE

Figure 2
Second-Order Simplex Designs

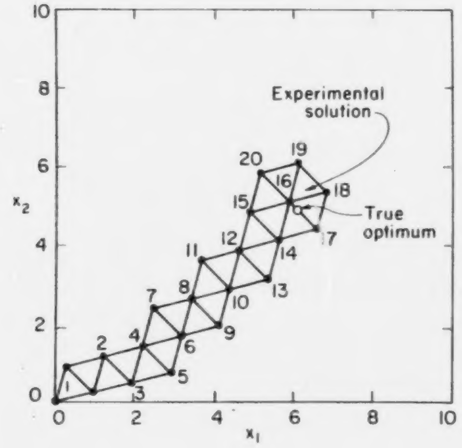


Figure 3
Sequential Simplex Search

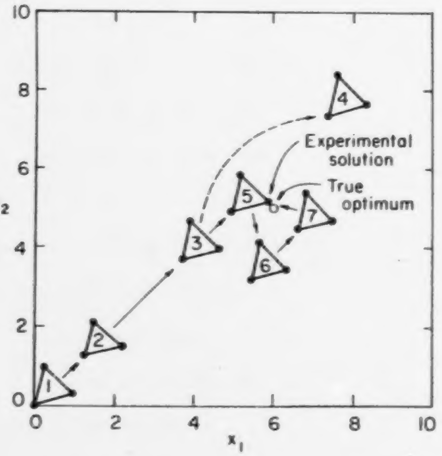


Figure 4
Accelerated Sequential Simplex Search

OPTIMIZATION OF SIMULATION EXPERIMENTS

R. E. Taylor

J. W. Schmidt

V. Chachra

Computing Center
and
Department of Industrial Engineering and Operations Research
Virginia Polytechnic Institute and State University
Blacksburg, Virginia

Abstract

The basic objectives of this paper are two-fold. The first objective is to illustrate the use of three multivariable optimization techniques as they are applied in an interactive fashion to the optimization of simulation experiments. The second and more important objective is to present the rationale behind the termination criterion for simulation experiments which is applicable to virtually any multivariable optimization procedure. The termination criterion is statistically based and includes cost factors prevalent for running the simulation as well as the potential savings from continued application of the search. The optimization techniques to be considered in the paper are

1. The sequential one factor-at-a-time technique as proposed by Friedman and Savage,
2. The pattern search method of Hooke and Jeeves, and
3. The successive quadratic approximation technique of Schmidt and Taylor.

It is shown that the termination criteria based upon economic and statistical considerations is most effective for simulation experiments.

Introduction

Digital simulation techniques for discrete systems have progressed rapidly over the past decade. The present development in digital simulation seems to be following two basic avenues. The first area of development is in special purpose computer languages for discrete systems simulation. The second area of development is in the statistical methodology related to the design of simulation experiments and analysis of results. In this paper we shall investigate a third possible avenue of development for discrete systems simulation, namely the use of multivariate optimization techniques for simulation experiments.

Often in performing the simulation analysis of a given system the objective is simply to obtain a measure of system effectiveness for some prescribed values of the decision variables. However, more frequently, the objective is to obtain the specific values of decision variables which will optimize the system effectiveness function. When this is in fact the objective, the problem can be addressed by a body of "multivariable optimization techniques". Indeed these techniques are not new. They have been in existence for many years and have been applied widely to problems of a deterministic nature [4], [18], [19].

The basic objectives of this paper are two fold: The first objective is to illustrate the application of three of these techniques to the optimization of simulation models. The second and more important objective is to present the rationale behind a termination criterion for simulation experiments which is applicable to virtually any multivariable optimization procedure. The termination criterion is statistically based and includes the cost factors prevalent in conducting the simulation analysis as well as potential saving from continued application of the search. The termination criterion will be shown to be effective for these stochastic problems.

The optimization techniques are employed in an interactive manner with the simulation model. The operation is such that particular values of the decision variables are specified by the optimization program to the simulation program. A measure of system effectiveness is determined through simulation which is returned to the optimization program. Based upon that value of the effectiveness function new values of the decision variables are determined and the process is repeated. At some point in this process a termination criterion will be met and the procedure will terminate. This facet will be discussed later.

The optimization techniques to be considered in this paper are:

1. The sequential one-factor-at-a-time technique as proposed by Friedman and Savage (7).
2. The pattern search method of Hooke and Jeeves (10).
3. Successive quadratic approximations of

Schmidt and Taylor (16).

Each of these techniques shall be discussed in detail in a later section of this paper. The search routines will be discussed in terms of a minimization problem. The model to which the techniques were applied is a stochastic inventory system which shall also be discussed in some detail in later sections.

Multivariate Search Procedures

All multivariable search procedures have essentially two basic objectives: (1) to obtain an improved value of the effectiveness function; (2) to provide information useful for locating future experiments where desirable values are likely to be found. The logical organization of a search procedure is such as to accomplish the aforementioned objectives through a three phase operation. The first phase sets the stage by making the initial observation(s) of the effectiveness function. From this initial phase can be determined the general direction of the search. The second phase of the search is characterized by rapid movement toward the optimal. During this phase the effectiveness function is examined through selective manipulation of the decision variables. The final phase of the search is perhaps the most important. This is known as the termination phase and the termination criterion plays a critical part in the overall procedure.

In general, the first phase of any search procedure is designed to "get things underway". For most practical examples this phase consists of the experimenter "arbitrarily" establishing the stating point. Many experimental statistical designs have been created to aid in this process. However, for the procedures discussed herein the stating point is chosen arbitrarily and to some extent the results to be derived from any of these procedures are dependent upon a "lucky" choice for the beginning point. If the experimenter fortunately chooses initial values of decision variables which are close to the optimum levels, money will be saved in achieving a relative optimum. If, on the other hand, luck is not with the experimenter and he selects initial levels which are far from the optimal values, then it likely will cost him more to achieve a relative optimum.

Once the initial experiment has been accomplished the information gained from that may be used to assist future experiments. The procedure of the particular search technique is then applied in an algorithmic fashion. The search procedures discussed herein all operate on the function in a systematic fashion, varying the decision variables in some prescribed manner. This phase of the overall operation is likely to consume the bulk of the activity of the search. As a result of this phase, the effectiveness function should be significantly improved. Later sections of this paper will describe in detail this phase of the operation.

The final phase of the search procedure is called the termination phase and specifies the conditions under which the search procedure will terminate. This phase is of great interest and is considered at length in this paper.

There are several characteristics of search procedures which will be mentioned here for purposes of description. They will not be explored in depth but should be taken into account when considering what technique to apply. These characteristics are listed below.

1. Total number of simulation replications required to obtain an optimum.
2. Ability to move on the response surface in several directions.
3. Ability to vary step length.
4. Ability to deal successfully with a large number of decision variables.
5. Termination criterion.

The Sequential One-Factor-at-a-Time Method

Sectioning or the one-at-a-time method proposed by Friedman and Savage (7) is one of the simplest optimum seeking techniques available and may be applied to functions of any number of decision variables. Suppose $y(x_1,x_2,\ldots,x_n)$ is a cost function to be minimized, where x_i, $i=1,2,\ldots,n$, are the decision variables. To apply the method of sectioning, the analyst fixes the values of the last n-1 variables and varies the first until a minimum, or at least near minimum, is found. Let x_1^o be the minimizing value of x_1 with associated cost $y(x_1^o,x_2,\ldots,x_n)$. The value of x_1 is now fixed at x_1^o, and x_2 is varied until its optimal value is determined, x_2^o. This procedure is repeated for all n decision variables. The entire process is repeated until values of the decision variables are found such that further change in any one of the variables will result in an increase in the value of the objective function.

The sectioning search may be effected in several ways. However, the initial step is always the same. All but one of the decision variables are given fixed values. Let these variables be $x_2,\ldots,x_n$. The initial value of the remaining variable, x_1, must now be set and the measure of effectiveness, $y(x_1,\ldots,x_n)$, evaluated. The initial search over x_1 usually involves changing x_1 in rather large increments. Let δ_{ij} be the jth increment chosen for the ith decision variable and let m be the number of increments for each variable, $j=1,2,\ldots,m$. Choosing δ_{11} relatively large allows the search to rapidly locate the general region of the optimum value of x_1, x_1^*, given the fixed values of the remaining variables. Let us arbitrarily assume that in searching over any decision variable, we first increase the value of the variable and if this does not prove fruitful we then decrease its value. Therefore, the first step in the search moves us to the point $(x_1+\delta_{11},x_2,\ldots,x_n)$. If $y(x_1+\delta_{11},x_2,\ldots,x_n) < y(x_1,\ldots,x_n)$, we must continue to increase x_1, next examining the measure of effectiveness at $(x_1+2\delta_{11},x_2,\ldots,x_n)$. This procedure is continued until a point $(x_1+M\delta_{11},x_2,\ldots,x_n)$ is found such that $y(x_1+M\delta_{11},x_2,\ldots,x_n) > y(x_1+(M-1)\delta_{11},x_2,\ldots$

$.,x_n)$. If the objective function is convex, x_1^* lies between $x_1+(M-2)\delta_{11}$ and $x_1+M\delta_{11}$.

If $y(x_1+\delta_{11},x_2,\ldots,x_n) > y(x_1,\ldots,x_n)$, a further increase in x_1 would not be warranted if the objective function is convex. Therefore, the next point evaluated would be $(x_1-\delta_{11},x_2,\ldots,x_n)$. If $y(x_1-\delta_{11},x_2,\ldots,x_n) > y(x_1,\ldots,x_n)$, then x_1^* is such that $x_1-\delta_{11} < x_1^* < x_1+\delta_{11}$. If $y(x_1-\delta_{11},x_2,\ldots,x_n) < y(x_1,\ldots,x_n)$, x_1 is further reduced until a point $(x_1-M\delta_{11},x_2,\ldots,x_n)$ is found such that $y(x_1-M\delta_{11},x_2,\ldots,x_n) > y(x_1-(M-1)\delta_{11},x_2,\ldots,x_n)$, in which case x_1^* is such that $x_1-M\delta_{11} < x_1^* < x_1-(M-2)\delta_{11}$.

Ignoring boundary constraints, the result of the initial search over x_1 is an interval of width $2\delta_{11}$, the center of which, x_1^o, is the best estimate of x_1^* thus far. At this point the analyst may choose to continue the search over x_1, keeping the remaining decision variables fixed at their previously established values. To accomplish this, the analyst chooses a new increment for x_1, δ_{12}, which is less than the initial increment. The starting point for this search is the center point of the interval about x_1^* which was obtained in the initial search, x_1^o. The procedure described for the initial search of x_1 is then repeated until a new value of x_1^o is derived. The entire process is repeated over and over again until x_1^o is bracketed by a sufficiently small interval. When the search over x_1 terminates, the search over x_2 begins, fixing x_1 at the last value of x_1^o derived and holding $x_3,\ldots,x_n$ at their initial values. The procedure for the search over x_2 is identical to that for x_1. After all n variables have been searched over once, the search returns to x_1 and starts the whole process over again. The search terminates when for every i

$$y(x_1^o,x_2^o,\ldots,x_i^o\pm\delta_{im},\ldots,x_n^o) >$$

$$y(x_1^o,x_2^o,\ldots,x_i^o,\ldots,x_n^o) \tag{1}$$

When the initial search over x_1 terminates, the analyst may choose to search over the remaining variables before refining the search over x_1. If this is the case, x_1 is fixed at the initial value of x_1^o, and the search over x_2 is conducted in increments δ_{21}. This process is repeated for all n variables. Here the search returns to x_1, again searching in increments δ_{11}. The search increment for any variable is not reduced until a point $(x_1^o,x_2^o,\ldots,x_n^o)$ is found such that for every i

$$y(x_1^o,x_2^o,\ldots,x_i^o\pm\delta_{i1},\ldots,x_n^o) >$$

$$y(x_1^o,x_2^o,\ldots,x_i^o,\ldots,x_n^o) \tag{2}$$

When this condition is achieved, the increments on all variables are reduced to δ_{i2}, $i=1,2,\ldots,n$, and the search over all decision variables is repeated until the termination criterion given is satisfied.

The Pattern Search Method

The philosophy underlying the pattern search technique is based upon the hopeful conjecture that any adjustments of the decision variables which have improved the effectiveness function during early experiments will be worth trying again. The technique begins from the starting point by moving in small steps. The steps grow with repeated success. Failure at any step length indicates that shorter steps are in order. If a change in direction is required, the technique will begin over again with a new pattern. The method is a ridge following technique and a pattern of moves can succeed only if it lies along a straight ridge. In the area of the optimal the steps become very small to avoid overlooking any promising direction. As before $y(\underline{x})$ is the value of the objective function evaluated at the point $\underline{x}$, previously defined as $(x_1,x_2,\ldots,x_n)$. The technique seeks an optimal in a series of cycles. One cycle differs from another basically in the step length employed for the decision variables.

In visualizing what is meant by a "pattern", it is helpful to think of an arrow, its base at one end and its head at the other. A cycle begins at a base point $\underline{b}_1$. At the beginning of a given cycle a step width δ_i is determined for each decision variable. Let $\underline{\delta}_i$ be the vector whose ith component is δ_i, the rest being zero. After evaluating $y(\underline{b}_1)$, $y(\underline{b}_1+\underline{\delta}_1)$ is evaluated. If the new point, $\underline{b}_1 + \underline{\delta}_1$, is better than the base point, this point is called the temporary head $\underline{t}_{11}$, where the first subscript indicates the pattern number under construction and the second subscript indicates the variable number most recently perturbed. If $\underline{b}_1 + \underline{\delta}_1$ is not as good as $\underline{b}_1$, $y(\underline{b}_1-\underline{\delta}_1)$ is evaluated. If this point is better than the base point it is denoted as the temporary head; otherwise $\underline{b}_1$ is designated as the temporary head. This process is repeated for each of the decision variables following the rule that a jth temporary head $(\underline{t}_{ij})$ is obtained from the preceding one, $\underline{t}_{i,j-1}$, as follows:

$$\underline{t}_{ij} = \begin{cases} \underline{t}_{i,j-1} + \underline{\delta}_j \text{ if } y(\underline{t}_{i,j-1} + \underline{\delta}_j) < y(\underline{t}_{i,j-1}) \\ \underline{t}_{i,j-1} - \underline{\delta}_j \text{ if } y(\underline{t}_{i,j-1} - \underline{\delta}_j) < y(\underline{t}_{i,j-1}) \\ \underline{t}_{i,j-1} \text{ if } y(\underline{t}_{i,j-1}) < \\ \qquad \min[y(\underline{t}_{i,j-1} + \underline{\delta}_j), y(\underline{t}_{i,j-1} - \underline{\delta}_j)] \end{cases} \quad (3)$$

Equation 3 covers all variables $j(1 \leq j \leq n)$ if the convention is adopted that

$$\underline{t}_{10} \equiv \underline{b}_1 .$$

When all decision variables have been perturbed, the last temporary head point, $\underline{t}_{1n}$ is designated as the second base point $\underline{b}_2$, i.e.,

$$\underline{t}_{1n} \equiv \underline{b}_2 .$$

The original base point in the cycle $\underline{b}_1$ and the newly determined base point, $\underline{b}_2$, establish the first pattern which is the arrow joining $\underline{b}_1$ to $\underline{b}_2$.

At this point in the procedure an acceleration step is initiated to establish the next temporary heading. Under the philosophy that if a similar exploration is conducted from $\underline{b}_2$

the results are likely to be the same, the local perturbations are ignored and the search is extended to a new temporary head $\underline{t}_{20}$ for the second pattern based at $\underline{b}_2$. The initial temporary head is given by

$$\begin{aligned} \underline{t}_{20} &\equiv \underline{b}_1 + 2(\underline{b}_2-\underline{b}_1) \\ &= \underline{b}_2 + \underline{b}_2 - \underline{b}_1 \\ &= 2\underline{b}_2 - \underline{b}_1 \quad . \end{aligned} \tag{4}$$

In other words, the arrow (representing the direction of the pattern) is extended from $\underline{b}_1$ to $\underline{b}_2$, immediately doubling its length. In line with terminology previously used, the double subscript on the temporary head $\underline{t}_{20}$ indicates the initiation of the second pattern with no local explorations yet performed. A local exploration is now carried out about $\underline{t}_{20}$ to correct the tenative second pattern, if necessary. The logical equations governing establishment of new temporary head $\underline{t}_{21},\underline{t}_{22},\ldots,\underline{t}_{2n}$ will be similar to Equation 3, the only difference being that the first subscript will be 2 instead of 1. If, after all variables have been perturbed, the last temporary head $\underline{t}_{2n}$ is better than $\underline{b}_2$ it is designated as the third base point $\underline{b}_3$.

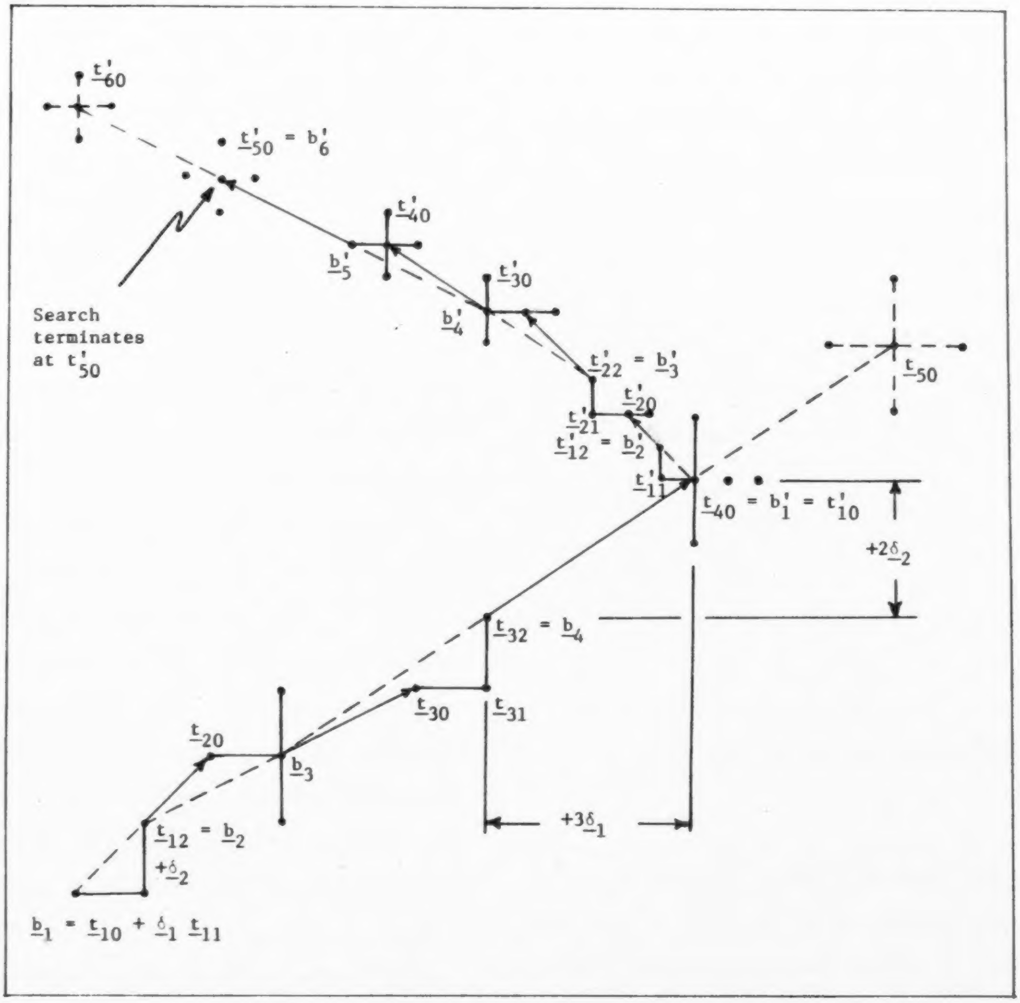
As before, a new temporary head $\underline{t}_{30}$ is established by extrapolating from $\underline{b}_2$ through $\underline{b}_3$, i.e.,

$$\underline{t}_{30} = 2\underline{b}_3 - \underline{b}_2 \ . \tag{5}$$

Repeated success in a given direction causes the pattern to grow and as long as this procedure improves the objective function it is continued.

If, however, the attempt to establish a new temporary head is unsuccessful the pattern is destroyed and perturbation of the independent variables is begun at the current δ values about the last successful base. If these perturbations are successful the pattern will again begin to grow and accelerate. If, on the other hand, perturbations about the last successful base are unsuccessful then the cycle is complete. A new cycle is begun by reducing the step size (elements of $\underline{\delta}_1$), and initiating perturbations about the base with the new step size. The termination criterion for the pattern search is normally couched in terms of step size used for perturbations. When the minimum step size is reached the technique is terminated.

Figure 1 illustrates the operation of the pattern search technique for a two variable case. In this example the search begins by proceeding in the positive direction for both decision variables. As the search proceeds successfully during cycle one the pattern continues to grow. At temporary heading $\underline{t}_{40}$ the search falters and is unable to make further improvements at the existing step width. At this point the step width is decreased and the second cycle begins. Note that during the second cycle (indicated by primes) the direction of the search completely changes. At point $\underline{t}'_{50}$ the search is unable to find further improvements. In that the step



t'60
t'50 = b'6
Search terminates at t'50
t'40
b'5
t'30
b'4
t'22 = b'3
t'20
t'21
t'12 = b'2
t'11
t40 = b'1 = t'10
t50
+2δ2
x2
t32 = b4
t30
t31
t20
b3
+3δ1
t12 = b2
+δ2
b1 = t10 + δ1 t11
x1

FIGURE 1: Operation of Pattern Search

width has reached its minimum value the procedure terminates. An excellent discussion of this technique can be found in Wilde and Beightler (19).

Successive Quadratic Approximation Method

The search by successive quadratic approximation is based upon the assumption that the objective function can be roughly approximated by a quadratic equation. The reliability of the approximation increases as the region of the optimal to which the approximation applies is reduced. Let $y(x_1,x_2,\ldots,x_n)$ be the objective function and $(x_1,x_2,\ldots,x_n)$ the decision variables. The approximating function, $\hat{y}(x_1,x_2,\ldots,x_n)$, can be expressed by

$$\hat{y}(x_1,x_2,\ldots,x_n) = b_o + \sum_{i=1}^{n} b_i x_i + \sum_{i=1}^{n} b_{i+n} x_i^2 . \tag{6}$$

Let m be the number of coefficients in the approximating expression. Therefore

$$m = 2n + 1 .$$

The approximating function given in Equation 6 may be augmented by the addition of terms such as $x_i x_j$ to improve the approximation. However, the authors have found Equation 6 satisfactory in most cases. The constants, b_i, which specify $\hat{y}(x_1,x_2,\ldots,x_n)$ are developed through the method of least squares. Therefore, $y(x_1,x_2,\ldots,x_n)$ must be evaluated at $k \geq m$ points $(x_{1j},x_{2j},\ldots,x_{nj})$, $j=1,2,\ldots,k$. In the context of this paper $y(x_1,x_2,\ldots,x_n)$ is evaluated through simulation although an appropriate mathematical model could be used for this purpose if it were available.

Having fit the approximating function to k points in the solution space, $\hat{y}(x_1,x_2,\ldots,x_n)$ is optimized through the classical methods of calculus. That is

$$\frac{\partial \hat{y}}{\partial x_i} = b_i + 2b_{i+n}\, x_i = 0 \tag{7}$$

and

$$x_i^* = -b_i/2b_{i+n} \tag{8}$$

The point $(x_1^*,x_2^*,\ldots,x_n^*)$ represents the initial estimate of the optimum for $y(x_1,x_2,\ldots,x_n)$ and is the next point at which $y(x_1,x_2,\ldots,x_n)$ is evaluated. Once $y(x_1,x_2,\ldots,x_n)$ has been evaluated at $(x_1^*,x_2^*,\ldots,x_n^*)$, the point $(x_{1j},x_{2j},\ldots,x_{nj})$ for which $y(x_1,x_2,\ldots,x_n)$ is least optimal, $j=1,2,\ldots,k$, is dropped from further consideration. Let the least optimal point be denoted by $(x_1',x_2',\ldots,x_n')$. Therefore the number of points in the analysis is still k, but $(x_1',x_2',\ldots,x_n')$ is replaced by $(x_1^*,x_2^*,\ldots,x_n^*)$. Again applying the method of least squares, $\hat{y}(x_1,x_2,\ldots,x_n)$ is fit to the new set of k points and the entire procedure is repeated.

As the search progresses the region of investigation of the solution space will generally contract about the optimal point, although this general contraction may be accompanied by periodic expansions. This variation is

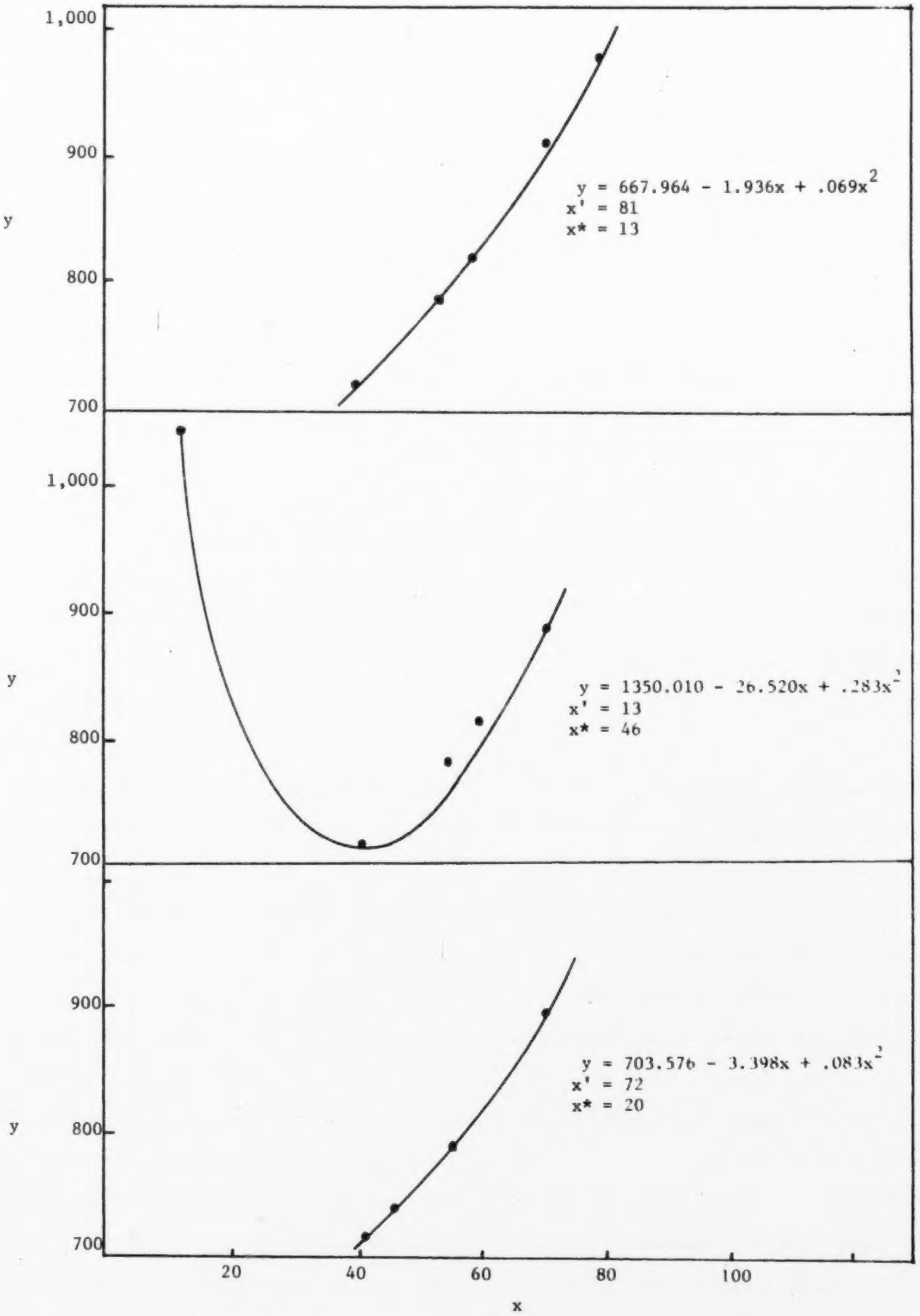


FIGURE 2: Application of Successive Quadratic Approximation

illustrated in Figure 2 where y is a function of one variable, x, and the objective of the search is minimization of y. Three successive iterations of the search are shown in which the region of the search varies from $41 \leq x \leq 81$ to $13 \leq x \leq 72$ to $41 \leq x \leq 72$.

The analyst may adopt a termination criterion of his own choice in using the search by successive quadratic approximation. For example, termination may be effected by specifying a fixed number of iterations. Another alternative is to terminate the search whenever the region of investigation is reduced to a sufficiently small neighborhood. However, these criteria are most effective when $y(x_1,x_2,\ldots,x_n)$ can be expressed in mathematical form.

Termination of the Search

When the system model to be optimized is a mathematical model, the search procedure usually terminates either after a fixed number of iterations or when the step size in the exploratory segment of the search has been reduced to a predefined minimum, although other termination criteria may also be used. The purpose of the application of a search procedure is to identify a point at or near the optimum for the mathematical model. To insure this, the termination criteria is usually defined in such a manner that the search continues well beyond the identification of an adequate approximation to the true optimum. Thus, there is normally wasted computer time resulting from excessive iterations. However, the cost of these extra iterations may not be expensive, since many mathematical models can be evaluated rapidly on a digital computer.

When the system to be optimized is modeled through simulation, the cost of evaluation of the model can be expensive. Thus excessive iteration of the search procedure may result in a situation where more is spent identifying the optimum or near optimum than was saved by finding the optimum. The cost of simulation arises from two sources. Let $(x_1,x_2,\ldots,x_n)$ be a vector of decision variables representing a point at which the system is to be evaluated in the course of the search and let $y(x_1,x_2,\ldots,x_n)$ be the corresponding expected cost of operation of the system. Since $y(x_1,x_2,\ldots,x_n)$ is to be evaluated through simulation, the value of $y(x_1,x_2,\ldots,x_n)$ can only be estimated. Let $\hat{y}$ be the estimate of $y(x_1,x_2,\ldots,x_n)$ obtained by one replication of the simulation. Then

$$\hat{y} = y(x_1,x_2,\ldots,x_n) + \varepsilon \qquad (9)$$

where ε is a random variable representing the error due to simulation with mean zero and variance σ^2. One replicate of the simulation at $(x_1,x_2,\ldots,x_n)$ alone may be expensive. In addition, depending upon the value of σ^2, one replicate may provide a poor estimate of $y(x_1,x_2,\ldots,x_n)$. To improve the estimate of $y(x_1,x_2,\ldots,x_n)$, we may replicate the simulation N times at $(x_1,x_2,\ldots,x_n)$. Let $\hat{y}_1,\hat{y}_2,\ldots,\hat{y}_N$ be the simulated values of the cost of operation of the system for replications 1,2,...,N. Then

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{N} \hat{y}_i \tag{10}$$

has mean $y(x_1,x_2,\ldots,x_n)$ and variance $\frac{\sigma^2}{N}$.

Let C_r be the cost of one replicate of the simulation, k the number of iterations of the search until termination, and r_i the number of replicates at the ith point evaluated in the search. Then the total cost of executing the search is:

$$\text{Cost of Search} = C_r \sum_{i=1}^{k} r_i . \tag{11}$$

Let $\bar{y}_1$ be the estimated cost of the system at the first point evaluated in the search and $\bar{y}_o$ the minimum estimated cost found in the course of the search. Then the savings in system cost achieved by the search is $\bar{y}_1 - \bar{y}_o$, and an attempt should be made to design the search such that

$$\bar{y}_1 - \bar{y}_o > C_r \sum_{i=1}^{k} r_i \tag{12}$$

Of course, there is no guarantee, prior to execution of the search, that the expression in Equation 12 will be satisfied. For example, if the starting point for the search is close to the optimum, large savings as a result of the search may not be possible. Therefore the termination criterion should be designed such that a condition of this type will be detected quickly and the search terminated.

Let $\bar{y}_1,\bar{y}_2,\ldots,\bar{y}_k$ be the costs of operation of the system as estimated through $r_1,r_2,\ldots,r_k$ replications of the simulation for the first k points evaluated in the course of the simulation. Let

$$z_1 = \bar{y}_1$$

and define $z_2,z_3,\ldots,z_p$, $p \leq k$ as a subset of $\bar{y}_1,\ldots,\bar{y}_k$ such that

$$z_j = \bar{y}_\ell$$

where ℓ is the smallest i for which

$$\bar{y}_i < z_{j-1} .$$

The definition for the z's given above holds for all optimization procedures which have a single point in solution at any time. In optimization procedures where more than one point is in solution at a given time, as in the sequential quadratic approximation method, a slightly different definition is necessary. If m is the number of points in solution at a given time, then define S_i as the set of m points in solution at the ith iteration. The set S_1 will consist of the first m points that are evaluated. Also define

$$z_1 = \bar{y}_1$$

then,

$$z_j = \bar{y}_\ell$$

where ℓ is the smallest index i for which

$$\bar{y}_i < \max\ [\bar{y} \mid \bar{y} \in S_{j-1}]$$

and S_j is the set of m points obtained from S_{j-1} by replacing the worst point in S_{j-1} by z_j.

To illustrate consider the sequence of values of $\bar{y}_i$ shown below for the first 15 iterations of a search. As shown, z_j is the value of the objective function, as estimated through simulation, at the point in the search at which the jth improvement in the estimated value of the objective function was observed.

Iteration i	Simulated Cost $\bar{y}_i$	j	z_j
1	1000	1	1000
2	1268		
3	1342		
4	987	2	987
5	1014		
6	824	3	824
7	915		
8	927		
9	887		
10	831		
11	807	4	807
12	801	5	801
13	835		
14	763	6	763
15	771		

At the jth improvement the loss resulting from the search up to that point is calculated and given by

$$L_j = z_j - z_1 + R \tag{13}$$

where R is the cost of the search up to and including the jth improvement. R is given by Equation 11 where k is the iteration at which the jth improvement occurred. A simple straight line of the form

$$L = b_o + b_1 x, \tag{14}$$

is then fit to the last $M \leq j$ improvements by the method of least squares. If the slope, b_1, of the straight line given in Equation 14 is significantly less than zero then there is reason to believe that continuation of the search may yield further savings. However, if the slope is greater than or equal to zero, then it is likely that the search should be discontinued in the sense that the maximum savings has already been achieved.

Since L_j is a random variable, the slope, b_1, is a random variable. To determine whether or not the slope is significantly less than zero, a t-test is conducted each time an improvement, z_j, is detected. If,

$$\frac{b_1}{S_{b_1}} < t_\alpha \left[\sum_{t=j-M+1}^{j} r_t \right] \tag{15}$$

where

r_t = the number of replications at the t^{th} improvement

then, the hypothesis

H_o: $b \geq 0$

is rejected, and the search continues. Otherwise the search is terminated.

To implement the termination criteria it is necessary to specify α and determine r_t and M. After each iteration of the search the number of replications at the next point is evaluated and the number of points, M, to which the straight line will be fit, if the t-test is conducted, are determined. However, these calculations require specification of a breakeven rate of return, ρ_b, on investment, a desirable rate of return, ρ_d, and the β error corresponding to the desirable rate of return. The logic incorporated into the search routine then calculates r_t and M such that the equations

$$P(\text{search continues} | b_1 = 0) = \alpha \qquad (16)$$

$$P(\text{search continues} | b_1 = \rho_b) = .5 \qquad (17)$$

$$P(\text{search continues} | b_1 = \rho_d) = 1-\beta \qquad (18)$$

are satisfied as nearly as possible. In any case the minimum values of M and r_t are never less than 2. However, the user may specify a minimum greater than 2 for either M or r_t or both.

The Model to Be Optimized

The function whose expected value is to be minimized is

$$y = 5. \sum_{i=1}^{5} [\frac{A_i B_i}{x_i} + \frac{C_i x_i}{2} (1 - \frac{A_i}{D_i})] + \varepsilon \qquad (19)$$

where

i	A_i	B_i	C_i	D_i
1	100	10	1	1000
2	200	20	4	1000
3	300	40	3	1000
4	400	100	5	1000
5	500	50	8	2000

and ε is distributed uniformly on the interval [-25, 25]. The minimum value of E(y) is approximately 7300 at (47, 50, 107, 163, 91). Each procedure began at (500, 500, 500, 500, 500). The value of y at this point is approximately 19820.70. The response surface for this model is a well behaved surface with a rather large "flat" region about the optimal. A sensitivity analysis performed about the optimal shows little response to changes in decision variables for a rather large area. The response surface is flat enough that random error can easily mask out true differences in response thereby possibly causing a premature termination of a search procedure.

Method of Operation

The model discussed in the previous paragraph was coded as a FORTRAN IV subprogram. Each optimization technique was coded as a FORTRAN IV main line program. The macro logic of the overall solution procedure is shown in Figure 3. The general method of operation is that the optimization program specifies values of the decision variables which are passed to the simulation model. The number of replications required at the next point is then determined. This determination is based upon desired confidence levels and the sample variance as discussed previously. After the simulation is complete the subprogram returns a particular value of the effectiveness function. At this time a statistical test is performed to determine if a significant improvement in the objective function has been achieved. Based upon a synthesis of this information, the search program then calculates a new set of decision variables which are passed to the simulation model, and the entire process is repeated. This procedure

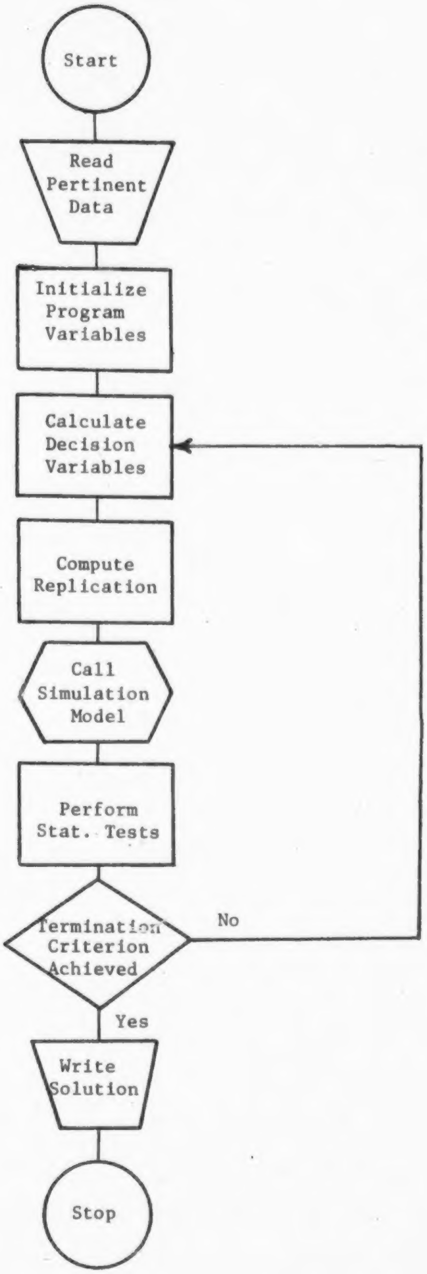

Start
Read Pertinent Data
Initialize Program Variables
Calculate Decision Variables
Compute Replication
Call Simulation Model
Perform Stat. Tests
Termination Criterion Achieved
No
Yes
Write Solution
Stop

FIGURE 3: Macrologic of Optimization Procedure

continues until the termination criterion is achieved at which time the process terminates.

Simulation Results

The simulation results obtained for the three optimization techniques are summarized in Tables 1-3. Values of $\alpha = 0.10$ and $\beta = 0.20$ were used for the statistical routines. The cost of replication was assumed to be $2.00. The results show that all three optimization techniques came fairly close to the actual optimal of 7300. This indicates that search procedures can successfully be employed to optimize simulation experiments. The optimal point, total number of replications and the simulated cost for each of the optimization techniques is presented in Table 4. No attempt is made to compare their relative performance.

The results also indicate that the termination criterion based on cost, α and β errors and the rates of return is effective. The number of iterations for the three search procedures ranges from 32 to 275 indicating that termination based only on the number of iterations would be grossly inadequate. Variations in the evaluation of the objective function are inherent in simulation experiments. Depending upon the magnitude of this variation, a step in the right direction may appear to be otherwise. A termination based only on the improvement between successive iterations, when encountering the above situation would terminate the search prematurely. To illustrate, termination under

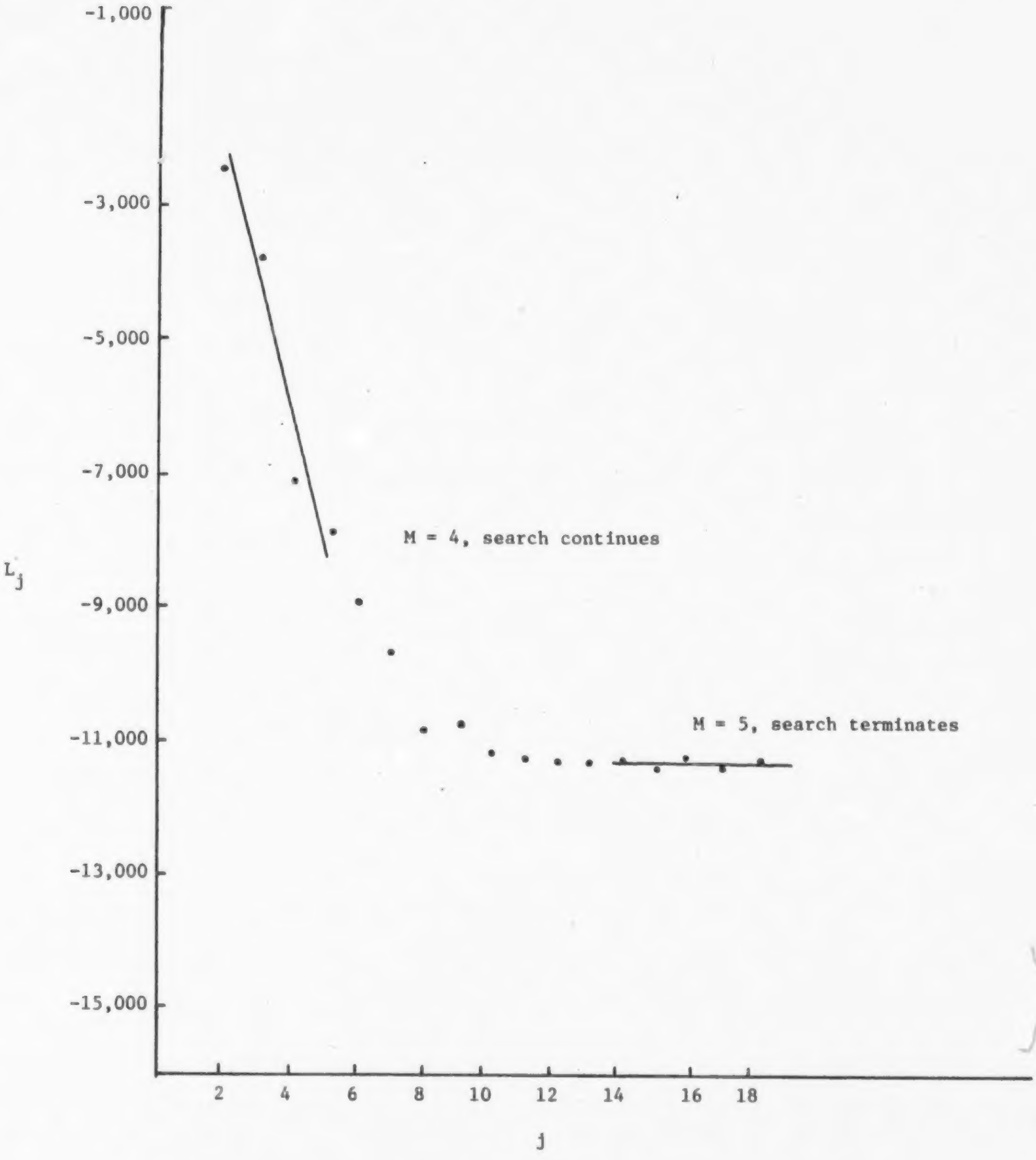


FIGURE 4: Losses L_j from Table 3 for a search where termination occurred at the 18th improvement

Iter #	Simulated Cost	Cumulative # of Replications	Cumulative Replication Cost	Improvement Point Number (j)	Value At Improved Point (z_j)	Loss L_j
1	19819.45	4	8.00	1	19819.45	-
2	20034.90	8	16.00	-	-	-
3	19594.81	12	24.00	2	19594.81	-200.64
4	19369.10	16	32.00	3	19369.10	-418.35
5	19164.61	21	42.00	4	19164.61	-612.84
6	18968.97	25	50.00	5	18968.97	-800.48
7	23701.50	28	56.00	-	-	-
8	19750.59	31	62.00	-	-	-
9	18175.44	34	68.00	6	18175.44	-1576.01
.						
.						
.						
29	7633.54	99	198.00	20	7633.54	-11988.46
42	7601.88	138	276.00	21	7601.88	-11942.12
45	7391.50	144	288.00	22	7391.50	-12140.50
50	7337.99	169	338.00	23	7337.99	-12143.46
73	7324.22	238	476.00	24	7324.22	-12019.23
81	7319.56	262	524.00	25	7319.56	-11975.89

TABLE 1: Simulation Results for One at a Time Search Applied to a Five Variable Inventory Problem

Iter #	Simulated Cost	Cumulative # of Replications	Cumulative Replication Cost	Improvement Point Number (j)	Value At Improved Point (z_j)	Loss L_j
1	19820.92	5	10.00	1	19820.92	-
2	19635.42	9	18.00	2	19635.42	-167.50
3	19010.94	13	26.00	3	19010.94	-783.97
4	18606.25	17	34.00	4	18606.25	-1180.67
5	18089.67	21	42.00	5	18089.67	-1689.24
6	16943.89	26	52.00	6	16943.89	-2825.01
7	14138.26	29	58.00	7	14138.26	-5624.66
8	13948.89	36	72.00	8	13948.89	-5800.01
9	13339.78	38	76.00	9	13339.78	-6405.13
10	12957.38	43	86.00	10	12957.38	-6777.53
11	12546.43	45	90.00	11	12546.43	-7184.49
12	11436.08	47	94.00	12	11436.08	-8290.84
13	7911.42	50	100.00	13	7911.42	-11809.50
14	7911.53	53	106.00	-	-	-
15	8074.42	56	112.00	-	-	-
.						
.						
21	7626.95	74	148.00	14	7626.95	-12045.96
.						
32	7596.78	129	258.00	15	7596.78	-11966.13

TABLE 2: Simulation Results for Pattern Search Applied to a Five Variable Inventory Problem

Iter #	Simulated Cost	Cumulative # of Replications	Cumulative Replication Cost	Improvement Point Number (j)	Value At Improved Point (z_j)	Loss L_j
1	19820.00	4	8.00	1	19820.00	-
2	17372.57	8	16.00	2	17372.57	-2431.43
3	25635.70	12	24.00	-	-	-
.						
5	16057.61	20	40.00	3	16057.61	-3722.39
.						
.						
.						
223	8193.09	456	912.00	10	8193.09	-10714.91
.						
228	7726.22	466	932.00	11	7726.22	-11161.78
.						
230	7635.86	470	940.00	12	7635.86	-11244.14
.						
237	7566.05	484	968.00	13	7566.05	-11285.95
.						
244	7525.02	498	996.00	14	7525.02	-11298.98
.						
250	7520.51	510	1020.00	15	7520.51	-11279.49
.						
251	7377.21	512	1024.00	16	7377.21	-11418.79
.						
265	7373.00	540	1080.00	17	7373.00	-11367.00
.						
269	7338.58	548	1096.00	18	7338.58	-11385.00
.						
275	7337.26	560	1120.00	19	7337.26	-11362.74

TABLE 3: Simulation Results for Sequential Quadratic Approximation Method Applied to a Five Variable Inventory Problem

Optimization Technique	Optimal Point					Total # of Replic.	Simulated Cost
	x_1	x_2	x_3	x_4	x_5		
One-at-a-Time	50.00	50.00	100.00	162.50	87.50	262	7320.11
Pattern Search	100.00	100.00	100.00	180.00	100.00	129	7596.78
Sequential Quadratic Approximation	58.24	47.25	120.99	175.19	92.67	560	7337.26
Theoretical Values	47	50	107	163	91		7300

TABLE 4: Summary of Simulation Results for Three Optimization Techniques Applied to a Five Variable Inventory Problem

this criterion would have occurred at iteration number 42 in Table 1. The termination criterion defined in this paper effectively overcomes the situation described above. Figure 4 illustrates the operation of the termination criterion. The data from the "Losses" column in Table 3 is used to plot Figure 4.

Conclusions

The results of this study demonstrate that search procedures may be effectively used in the optimization of simulation experiments. More importantly, a termination criterion based on cost of replications, α and β errors, minimum rate of return and desired rate of return is proposed. It is found that the termination criterion based on these economic and statistical considerations is effective for simulation experiments. This termination criterion may be used for any search procedure applied to stochastic systems.

Bibliography

1. Box, G. E. P., "The Exploration and Exploitation of Response Surfaces: Some General Considerations and Examples", Biometrics, 10, 1954.
2. Box, G. E. P., and Wilson, K. B., "On the Experimental Attainment of Optimum Conditions", Journal of the Royal Statistical Society, Vol. 13, No. 1, 1951.
3. Box, M. J., "A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems", Computer Journal, Vol. 9, No. 1, May 1966.
4. Brooks, Samuel H., "A Comparison of Maximum-Seeking Methods", Operations Research Journal, Vol. 7, 1959.
5. Draper, Norman R., "Ridge Analysis of Response Surfaces", Technometrics, Vol. 5, No. 4, November 1963.
6. Fletcher, R., and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization", Computer Journal, Vol. 6, No. 2, July 1963.
7. Friedman, Milton, and Savage, L. J., Chapter 13, "Planning Experiments Seeking Maxima", Techniques of Statistical Analysis, McGraw-Hill, New York, N. Y., 1947.
8. Graybill, Franklin A., An Introduction to Linear Statistical Models, Vol. 1, McGraw-Hill, New York, N. Y., 1961.
9. Gue, Ronald L., and Thomas, Michael E., Mathematical Methods in Operations Research, Macmillan, London, 1968.
10. Hooke, R., and Jeeves, T. A., "Direct Search Solution of Numerical and Statistical Problems", J. Assn. Comp. Mach., Vol. 8, No. 2, April 1961.
11. Kiefer, J., and Wolfowitz, J., "Stochastic Estimation of the Maximum of a Regression Function", Annals of Mathematical Statistics, 22, 1952.
12. Montgomery, Douglas Carter, "Some Techniques for Determining Machine Center Capacities in a Job-Shop", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1969.
13. Myers, R. H., Response Surface Methodology, Allyn and Bacon, Inc., Boston, Massachusetts, 1971.
14. Sasser, W. Earl, Burdick, Donald S., Graham, Daniel A., and Naylor, Thomas H., "The Application of Sequential Sampling to Simulation: An Example Inventory Model", Communications of the ACM, 1970.
15. Schmidt, J. W., and Taylor, R. E., Simulation and Analysis of Industrial Systems, Richard D. Irwin, Homewood, Illinois, 1970.
16. Schmidt, J. W., and Taylor, R. E., "System Optimization Through Simulation", Simulation, Vol. 18, No. 2, February 1972.
17. Smith, James Ross, "The Method of Successive Quadratic Approximations With Application to Simulation", Ph.D. Dissertation, Virginia Polytechnic Institute and State University, 1971.
18. Spendley, W., Hext, G. R., and Himsworth, F. R., "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation", Technometrics, November 1962.

19. Wilde, Douglass J., and Beightler, Charles, S., Foundations of Optimization, Prentice-Hall, Englewood Cliffs, N. J., 1967.

20. Wine, R. Lowell, Statistics for Scientists and Engineers, Prentice-Hall, Englewood Cliffs, N. J., 1964.

A NEW APPROACH TO SIMULATING STABLE STOCHASTIC SYSTEMS*

Michael A. Crane and Donald L. Iglehart
Control Analysis Corporation and Stanford University

Abstract

A technique is introduced for analyzing simulations of stochastic systems in steady-state. Confidence intervals are obtained for a general function of the steady-state distribution.

1. INTRODUCTION

The principal goal of most simulations of stable stochastic systems is to estimate properties of the stationary or steady-state behavior of the system. Two of the major problems in such simulations are the statistical dependence between successive observations and the inability of the simulator to begin the system in the steady-state. The first problem has necessitated using methods of time series analysis rather than classical statistics. The second has inspired many simulators to let the system run for a sufficient length of time so that the initial transient wears off and a steady-state condition obtains. This procedure, of course, requires a judgement on how long to let the system run before making observations.

For many stochastic systems being simulated it is possible to find a random grouping of observations which produces independent identically distributed (i.i.d.) blocks from the start of the simulation. This grouping then enables the simulator to avoid the two problems mentioned above. He has at his disposal the methods of classical statistical analysis such as confidence intervals, hypothesis testing, regression, and sequential estimation since the observations are now i.i.d. Furthermore, information that is useful in estimating the steady-state behavior of the system can be collected from scratch thus eliminating the problem of the initial transient.

The key requirement for obtaining these i.i.d. blocks is that the system being simulated return to a single state infinitely often and that the mean time between such returns is finite. This requirement will be met for many, but not all, stable systems that might be simulated.

In this paper we shall illustrate the main ideas of this approach as applied to Markov chains, in both discrete and continuous time, and to the GI/G/1 queue. The results will only be sketched here as the complete details are available in [1], [2], and [3]. This paper is organized as follows. Section 2 summarizes the

---

* This research was sponsored by Office of Naval Research contract N00014-72-C-0086 [NR-047-106] and prepared for delivery at the 1973 Winter Simulation Conference, January 17-19, 1973, San Francisco.

probabilistic structure of Markov chains with an eye toward using these results in carrying out a simulation. Section 3 does the same for the GI/G/1 queue. In Section 4 a statistical confidence interval is stated for the ratio of two means. Numerical illustrations of this method are given in Section 5 for the repairman problem and the M/M/1 queue. The reader who is only interested in the results and not the underlying theory can turn directly to Section 5 with little loss of continuity.

2. MARKOV CHAINS

Suppose we are interested in simulating a stochastic system evolving as a Markov chain (M.c.). Let $\{X_n : n \geq 0\}$ be a discrete M.c. defined on a probability triple $(\Omega, \mathfrak{F}, P)$ with discrete state space $I = \{0, 1, 2, \cdots\}$. Everything we do here can be carried over to the case of I finite. Assume that this M.c. is known to be irreducible, aperiodic, and positive recurrent. Under these conditions there will exist a unique stationary distribution, $\{\pi_i : i \in I\}$, for the M.c.

Select now a fixed state of the M.c. which we shall take for convenience to be the state 0. Now set $X_0 = 0$ with probability one; that is, we shall always begin our M.c. in the 0 state. Since the M.c. is assumed to be positive recurrent, there exists an infinite sequence of random time epochs $\{\beta_i : i \geq 0\}$ such that $X_{\beta_i} = 0$ with probability one. Thus the epochs β_i are the successive times the process returns to 0. We shall speak of the integers $\{\beta_{k-1} + 1, \cdots, \beta_k\}$ as constituting the k^{th} cycle of the M.c. Let $\alpha_i = \beta_i - \beta_{i-1}$, $i \geq 1$ and for $k \geq 1$ form the random vectors

$$\underset{\sim}{V}_k = \{\alpha_k, X_{\beta_{k-1}+1}, \cdots, X_{\beta_k}\} .$$

As a consequence of the fact that the random variables (r.v.'s) $\{\beta_k : k \geq 1\}$ are optional and finite with probability one it is possible to show the following results.

PROPOSITION 1. The random vectors $\{\underset{\sim}{V}_k : k \geq 1\}$ are independent and identically distributed.

This proposition lies at the heart of our method of analyzing simulations.

Now let f be a function from I to $(-\infty, +\infty)$ and suppose the object of our simulation is to estimate $\sum_{j \in I} f(j)\, \pi_j$, the stationary expected value of f. Define new r.v.'s

$$Y_k = \sum_{j=\beta_{k-1}}^{\beta_k - 1} f(X_j), \qquad k \geq 1 .$$

As an immediate corollary of Proposition 1 we state

COROLLARY 1. The sequences $\{\alpha_k : k \geq 1\}$ and $\{Y_k : k \geq 1\}$ are independent and identically distributed.

The second important result is

PROPOSITION 2. If $\sum_{j\in I} |f(j)|\pi_j < \infty$, then the

$$\sum_{j\in I} f(j)\,\pi_j = E\{Y_k\}/E\{\alpha_k\}\ .$$

Corollary 1 and Proposition 2 form the basis for our method. We mention in passing that all the results of this section carry over to the case where $\{X_n : n \geq 0\}$ is a Markov process with a general state space E, a single ergodic set and no cyclically moving sets, provided there exists a point (singleton set) to which X_n returns infinitely often with probability one and for which the expected length of the cycles is finite.

Suppose now we are interested in simulating a continuous time M.c. Let $\{X(t) : t \geq 0\}$ be a continuous time M.c. defined on a probability triple $(\Omega, \mathcal{F}, P)$ and having discrete state space $I = \{0, 1, 2, \cdots\}$ and standard transition matrix $\{p_{ij}(t) : t \geq 0, i,j \in I\}$. Again assume that the M.c. is irreducible and positive recurrent. As in the discrete case, there exists a unique stationary distribution, $\{\pi_i : i \in I\}$, of the M.c. Also the $\lim_{t\to\infty} p_{ij}(t) = \pi_j$, for all $i,j \in I$.

Now set $X(0) = 0$ with probability one. Since the state 0 will be entered an infinite number of times as a consequence of our assumption of positive recurrence, we can define $\rho_i (i \geq 1)$ to be the length of the i^{th} visit to state 0. Then let $\beta_0 = 0$, and

$$\beta_i = \inf\{t > \rho_i + \beta_{i-1} : X(t) = 0\},\quad i \geq 1\ .$$

Thus β_i is the time of i^{th} return to 0. If we let $\alpha_i = \beta_i - \beta_{i-1}$, $i \geq 1$, then α_i is the length of the i^{th} cycle from the state 0 and plays the same role as in the discrete time case.

While the technical details for the continuous case are much harder than the discrete case, the intuitive ideas are the same. Hence for this discussion we shall keep the details brief. Let f be a mapping from I to $(-\infty, +\infty)$ and define the r.v.'s

$$Y_k = \int_{\beta_{k-1}}^{\beta_k} f[X(s)]ds,\quad k \geq 1\ .$$

For this continuous time M.c. Proposition 2 and Corollary 1 continue to hold. These results provide the basis for analyzing simulations of continuous time M.c.'s.

3. QUEUES

Consider now a GI/G/1 queueing system in which the 0^{th} customer arrives at time $t_0 = 0$, finds a free server, and experiences a service time v_0. The n^{th} customer arrives at time t_n and experiences a service time v_n. Let the interarrival times $t_n - t_{n-1} = u_n$, $n \geq 1$. Assume that the two sequences $\{v_n : n \geq 0\}$ and $\{u_n : n \geq 1\}$ each consist of i.i.d. r.v.'s and are themselves independent. Let $E\{u_n\} = \lambda^{-1}$, $E\{v_n\} = \mu^{-1}$ and $\rho = \lambda/\mu$ where $0 < \lambda,\ \mu < \infty$. Thus $\mu(\lambda)$ has the interpretation of the mean service (arrival) rate. The parameter ρ is called the traffic intensity and is the natural measure of congestion for this system. We shall assume that $\rho < 1$, a necessary and sufficient condition for the system to be stable.

The principal system characteristics of interest are $Q(t)$, the number of customers in the system at time t; W_n, the waiting time (time for arrival to commencement of service) of the n^{th} customer; $W(t)$, the work load facing the server at time t; $B(t)$, the amount of time in the interval $[0,t]$ that the server is busy; and $D(t)$, the total number of customers who have been served and have departed from the system in $[0,t]$.

Here we shall review the basic structure of the GI/G/1 queue relevant to our simulation study. Using the notation of optional r.v.'s, it can be shown that there exists a sequence of r.v.'s $\{\beta_k : k \geq 0\}$ such that $\beta_0 = 0$, $\beta_k < \beta_{k+1}$, and $W_{\beta_k} = 0$ with probability one. In other words, the customers numbered β_k are those lucky fellows who arrive to find a free server and experience no waiting in the queue. The fact that there exists an infinite number of such customers is a direct consequence of the assumption that $\rho < 1$. The time axis $R_+^1 = [0,\infty)$ can be divided into alternating intervals during which the server is busy, idle, busy, etc. We call these intervals busy periods (b.p.'s) and idle periods (i.p.'s). An i.p. plus the proceding b.p. is called a busy cycle (b.c.). If we let $\alpha_k = \beta_k - \beta_{k-1}$, $k \geq 1$, then α_k represents the number of customers served in the k^{th} busy period (b.p.) and they are numbered $\{\beta_{k-1}, \beta_{k-1} + , \cdots , \beta_k -1\}$. The sequence $\{\beta_k : k \geq 1\}$ plays the same role here as in Section 2 on M.c.'s.

Next define the random vectors $\underset{\sim}{X}_k = (v_{k-1}, u_k)$ and $\underset{\sim}{Y}_k = \{\alpha_k, \underset{\sim}{X}_{\alpha_{k-1}+1}, \cdots, \underset{\sim}{X}_{\beta_k}\}$, $k \geq 1$. Observe that the vector $\underset{\sim}{Y}_1 = \{\alpha_1, \underset{\sim}{X}_1, \cdots , \underset{\sim}{X}_{\alpha_1}\}$ includes all the data required to completely construct the behavior of the system in the first b.p. Let f be a measurable function from $[0,\infty)$ to $(-\infty,\infty)$ and set

$$Y_k = \sum_{j=\beta_{k-1}}^{\beta_k - 1} f(W_j), \qquad k \geq 1.$$

Then Proposition 1 and Corollary 1 continue to hold. Hence we have the intuitively plausible conclusion that comparable r.v.'s in different b.p.'s are i.i.d. However, Proposition 2 must be replaced by

PROPOSITION 3. *If* $E\{|f(W)|\} < \infty$, *then the*

$$E\{f(W)\} = E\{Y_k\}/E\{\alpha_k\} .$$

where W *is the stationary waiting time.*

In addition to obtaining results for $E\{f(W)\}$ we can also handle the expected value of the stationary queue length and virtual waiting time, length of a b.p., b.c., or i.p. Furthermore, this technique can be extended to the queue GI/G/s, $s > 1$; see [1].

4. CONFIDENCE INTERVALS

From Propositions 2 and 3 we are confronted with the need to produce confidence intervals for the ratio of two means. Suppose we observe i.i.d. random (column) vectors $\underset{\sim}{X}_1, \underset{\sim}{X}_2, \cdots, \underset{\sim}{X}_n$, where $\underset{\sim}{X}_k = (Y_k, \alpha_k)$, and assume that $E\{\underset{\sim}{X}_1\} = \underset{\sim}{\mu} = (\mu_1, \mu_2)$ with $\mu_2 \neq 0$. Let the positive definite covariance matrix of $\underset{\sim}{X}_1$ be $\underset{\sim}{\Sigma}$ with elements given by

$$\Sigma = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} .$$

Let $\nu = \mu_1/\mu_2$. Our goal is to form a confidence interval for ν based on the observations $\{\underset{\sim}{X}_k : 1 \leq k \leq n\}$ where n is large. This problem was treated by ROY and POTTHOFF (1958) for the case of bivariate normal random vectors.

Let the sample mean of the n observations $\underset{\sim}{X}_1, \cdots, \underset{\sim}{X}_n$ be denoted by

$$\bar{\underset{\sim}{X}}(n) = \frac{1}{n} \sum_{i=1}^{n} \underset{\sim}{X}_i = \begin{pmatrix} \bar{Y}(n) \\ \bar{\alpha}(n) \end{pmatrix}$$

and the sample covariance matrix by

$$\underset{\sim}{S}(n) = \frac{1}{n-1} \sum_{i=1}^{n} (\underset{\sim}{X}_i - \bar{\underset{\sim}{X}}(n)) (\underset{\sim}{X}_i - \bar{\underset{\sim}{X}}(n))'$$

with elements

$$\underset{\sim}{S}(n) = \begin{pmatrix} s_{11}(n) & s_{12}(n) \\ s_{12}(n) & s_{22}(n) \end{pmatrix} .$$

Next let $Z_k = Y_k - \alpha\nu_k$, $k = 1, 2, \cdots, n$ and let $\bar{Z}(n) = \frac{1}{n} \sum_{k=1}^{n} Z_k$. Observe that the $E\{Z_k\} = 0$ and the $\sigma^2\{Z_k\} = \sigma^2 = \sigma_{11} - 2\nu\sigma_{12} + \nu^2\sigma_{22}$. The idea of introducing the Z_k's is due to ROY and POTTHOFF (1958) and is the key to the confidence interval we obtain. Let $z_\gamma = \Phi^{-1}(\gamma)$, where

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2} du .$$

Using the central limit theorem for sums of i.i.d. random variables and the strong law of large numbers we can show that for $0 < \gamma < 1$ and large n the random interval

$$\left[\frac{(\bar{Y}\bar{\alpha} - hs_{12}) - D^{1/2}}{\bar{\alpha}^2 - hs_{22}} , \frac{(\bar{Y}\bar{\alpha} - hs_{12}) + D^{1/2}}{\bar{\alpha}^2 - hs_{22}} \right] \quad (1)$$

surrounds the parameter ν with probability approximately $1-\gamma$, where

$$D = (\bar{Y}\bar{\alpha} - hs_{12})^2 - (\bar{Y}^2 - hs_{11}) (\bar{\alpha}^2 - hs_{22})$$

and

$$h = z^2_{1-\gamma/2}/n .$$

5. NUMERICAL EXAMPLES

We illustrate our methods with simulations of two different models. The first is the classical repairman problem, and the second is the M/M/1 queue. The theoretical results for these models are well known and provide a basis for comparison.

The repairman problem is a continuous time Markov chain that can be described as follows. We have $M + N$ identical pieces of equipment which have an exponential failure time with failure rate λ. At most N of these units operate at one time, the other M units being thought of as spares. When a unit fails, it is sent to a repair facility consisting of S repairmen (servers) having exponential repair rate μ. Let $X(t)$ denote the number of failed units undergoing or waiting for service at the repair facility at time t. With the above assumptions $\{X(t) : t \geq 0\}$ is a birth-death process, a special type of the continuous time M.c. discussed in Section 2.

Let X be a discrete random variable having the stationary distribution $\{\pi_i : i = 0, \cdots, M + N\}$ of the M.c. In other words, X is the random variable to which $X(t)$ converges in distribution. We simulated the repairman problem in order to estimate $E\{f(X)\}$ for various choices of the function f.

Now let $X(0) = 0$. In order to analyze the simulation recall that the process returns to the state 0 infinitely often, and α_k is the length of the k^{th} cycle from the state 0. As in Section 2, let Y_k be the integral of $f[X(t)]$ over the k^{th} cycle. From Proposition 2 and Corollary 1, we know that the random vectors $\{(Y_k,\alpha_k)\ k \geq 1\}$ are i.i.d. and that $E\{f(X)\} = E\{Y_k\}/E\{\alpha_k\}$. We may thus obtain a confidence interval for $E\{f(X)\}$ by simulating the system for a fixed number n cycles and applying the method of Section 4. In particular, let $\bar{Y}$ and $\bar{\alpha}$ denote respectively the sample means for Y_k and α_k in n observations, let s_{11} and s_{22} denote the sample variances, and let s_{12} denote the sample covariance between Y_k and α_k. A $100(1-\gamma)\%$ confidence interval for $E\{f(X)\}$ is then given by equation (1) of Section 4.

To illustrate, we consider seven choices for the function f:

i) $f_1(i) = i, \quad i = 0, \cdots, M + N \quad ;$

ii) $f_2(i) = i^2, \quad i = 0, \cdots, M + N \quad ;$

iii) $f_3(i) = \begin{cases} 0, & i = 0, \cdots, M \quad , \\ 1, & i = M + 1, \cdots, M + N \ ; \end{cases}$

iv) $f_4(i) = \begin{cases} 0, & i = 0, \cdots, S \quad , \\ 1, & i = S + 1, \cdots, M + N ; \end{cases}$

v) $f_5(i) = \begin{cases} S-i, & i = 0, \cdots, S \quad , \\ 0, & i = S + 1, \cdots, M + N ; \end{cases}$

vi) $f_6(i) = \begin{cases} 0, & i = 0 \quad , \\ 1, & i = 1, \cdots, M + N \quad ; \end{cases}$

vii) $f_7(i) = \begin{cases} 1, & i = 0 \quad , \\ 0 & i = 1, \cdots, M + N \quad . \end{cases}$

These functions allow us to estimate, respectively, the expected value of X, the second moment of X, the probability that X exceeds M (insufficient spares), the probability that X exceeds S (positive queue length), the expected number of idle servers, the probability that X exceeds zero (at least one server busy), and the probability that X equals zero (all servers idle). There are of course many other functions which would yield useful estimates of the steady-state behavior.

Table 1 shows 90% confidence intervals obtained after a run length of 300 cycles from the state $X(0) = 0$. The parameter settings used for this run were $N = 10$ operating units, $M = 4$ spares, $S = 3$ servers, failure rate $\lambda = 1$, and repair rate $\mu = 4$. Table 2 shows estimates for $E\{X\}$ in ten replications of the simulation.

TABLE 1

SIMULATION RESULTS FOR THE REPAIRMAN PROBLEM

Parameter	Theoretical Value	Point Estimate	90% Confidence Interval
$E\{f_1(X)\} = E\{X\}$	3.471	3.406	[3.205, 3.607]
$E\{f_2(X)\} = E\{X^2\}$	17.278	16.844	[15.094, 18.594]
$E\{f_3(X)\} = P\{X > M\}$	.306	.294	[.206, .328]
$E\{f_4(X)\} = P\{X > S\}$	.438	.429	[.393, .465]
$E\{f_5(X)\} = E\{[S - X]^+\}$	.678	.705	[.637, .773]
$E\{f_6(X)\} = P\{X > 0\}$	.939	.930	[.919, .942]
$E\{f_7(X)\} = P\{X = 0\}$	.061	.070	[.058, .081]

TABLE 2

ESTIMATES FOR $E\{X\}$ IN TEN SIMULATION REPLICATIONS OF THE REPAIRMAN PROBLEM

Replication	Point Estimate	Confidence Interval
1	3.406	[3.205, 3.607]
2	3.386	[3.221, 3.551]
3	3.384	[3.196, 3.571]
4	3.440	[3.260, 3.620]
5	3.234	[3.047, 3.420]
6	3.542	[3.373, 3.712]
7	3.433	[3.246, 3.620]
8	3.382	[3.163, 3.600]
9	3.380	[3.213, 3.548]
10	3.415	[3.234, 3.596]
Average	3.400	[3.216, 3.585]
Average length		0.369

Theoretical value of $E\{X\} = 3.471$

Our second example is the M/M/1 queue. We have Poisson arrivals, exponential service, and a single server. Although the queue length process is a birth-death process and could be treated like the repairman problem, we focus our attention here on the sequence of customer waiting times $\{W_n : n \geq 0\}$. Recalling the discussion of Section 3, the process returns to the state W_0 infinitely often, and the time intervals between returns define busy cycles (b.c.'s). Letting f be a function on the state space, letting Y_k be the sum of $f(W_n)$ over the k^{th} b.c. and letting α_k be the number of customers served in the k^{th} b.c., we once again have $E\{f(W)\} = E\{Y_k\}/E\{\alpha_k\}$, where W is the stationary waiting time and the random vectors $\{(Y_k, \alpha_k)\ k \geq 1\}$ are i.i.d. We may thus proceed exactly as before to obtain confidence intervals for $E\{f(W)\}$.

Table 3 shows 90% confidence intervals in ten replications of the queueing simulation, each consisting of 2000 busy cycles. For these runs, the customer arrival rate was assumed to be 5 and the service rate 10 so that $\rho = .5$. We consider only the function $f(W) = W$, although there are many other interesting possibilities.

TABLE 3

ESTIMATES FOR $E\{W\}$ IN TEN SIMULATION REPLICATIONS OF THE M/M/1 QUEUE

Replication	Point Estimate	Confidence Interval
1	0.110	[.096, .123]
2	0.091	[.080, .102]
3	0.095	[.084, .105]
4	0.111	[.087, .133]
5	0.096	[.083, .109]
6	0.100	[.087, .112]
7	0.092	[.081, .103]
8	0.099	[.084, .114]
9	0.096	[.082, .109]
10	0.090	[.078, .102]
Average	0.098	[.084, .111]
Average length		.027

Theoretical Value of $E\{W\} = .100$

REFERENCES

[1] CRANE, M.A. and IGLEHART, D.L. (1972). A new approach to simulating stable stochastic systems, I: General Multi-server queues. Technical Report No. 86-1, Control Analysis Corporation, Palo Alto, California.

[2] CRANE, M.A. and IGLEHART, D.L. (1972). Confidence intervals for the ratio of two means with application to simulations. Technical Report No. 86-2, Control Analysis Corporation, Palo Alto, California.

[3] CRANE, M.A. and IGLEHART, D.L. (1972). A new approach to simulating stable stochastic systems, II: Markov chains. Technical Report No. 86-3, Control Analysis Corporation, Palo Alto, California.

[4] ROY, S.N. and POTTHOFF, R.F. (1958). Confidence bounds on vector analogues of the "ratio of means" and the "ratio" of variances" for two correlated normal variates and some associated tests. Ann. Math. Statist. **29**, 829-841.

Session 6: Manufacturing Applications
Chairman: John W. O'Leary, Western Electric Company

The objective of this session is to bring together a wide range of new concepts in manufacturing simulation to stimulate discussion and interest. The concepts involve the strategic and tactical decisions that are made during the course of a successful application in the manufacturing environment. Strategic aspects revolve around the selling of results to management. Tactical considerations include the scope of modeling detail and in the simulation model. The over-all goal of the papers and discussion is to provide a systematic foundation for analysts to draw from during the development of practical applications. Tactical considerations are concerned with the step-by-step abstraction of the simulation model from the real-world environment, and include setting the level of detail with the model as well as choosing an appropriate language.

Papers

"Using an Extended Version of GERT to Simulate Priority and Assignment Rules in a Labor Limited Job Shop System"
M. J. Maggard, University of Texas; W. G. Lesso, University of Texas;
G. L. Hogg, University of Illinois; and D. T. Phillips, Purdue University

"Evaluation Job Shop Simulation Results"
Carter L. Franklin II, University of Pennsylvania

"Simulation Applied to a Manufacturing Expansion Problem Area"
J. Douglas DeMaire, Olin Corporation

"Cycle-Time Simulation for a Multiproduct Manufacturing Facility"
M. M. Patel, J. M. Panchal, and M. T. Coughlin,
IBM Corporation

Discussant: Willaim L. Berry, Purdue University

Using an Extended Version of GERT
To Simulate Priority and Assignment Rules
In a Labor Limited Jobshop System

Michael J. Maggard
Department of Management
The University of Texas at Austin

William G. Lesso
Operations Research Group
Department of Mechanical Engineering
The University of Texas at Austin

Gary L. Hogg
Department of Mechanical and Industrial Engineering
University of Illinois

Don T. Phillips
Center for Large-Scale Systems and
Department of Industrial Engineering
Purdue University

ABSTRACT

In the literature, prior to 1965, most research on jobshop systems was on machine limited queueing systems. More recently this research has been directed toward labor and machine limited queueing systems. In this direction, the authors have developed and implemented an extended version of GERT called GERTS III QR (a GERTS model able to handle queueing systems with resource limitations). This model is further refined to handle both homogeneous and heterogeneous classes of labor.

This paper describes the GERTS III QR model and gives an illustration of its application. The example is a jobshop system with service centers in parallel. Three alternative priority and assignment rules, first-come-first-served, random and shortest operation time are evaluated.

The vast majority of the literature on jobshops has been addressed to the machine limited queueing systems. Recently, some analyses of actual jobshops suggest that machinery may not be the critical item but that available labor and its relative efficiency at various machine centers may be the limiting factor, (4).

With the introduction of the human element into this type of system, another dimension is added to the decision-making process. With a strict machine limited system, the problem was to determine "good" machine loading rules. Conway et al (3), Nanot (12) and others have developed such strategies. When the system is labor limited, however, the problem becomes more complex, i.e. one must also specify labor assignment rules.

Complete Labor Assignment Procedure. The complete labor assignment procedure includes the queue priority rule and its related labor assignment rule. In this paper, these systems are designated as DRC systems, e.g. Dual Resource Constrained systems. In DRC systems, moreover, the labor class may be either homogeneous or heterogeneous in nature. In homogeneous systems all laborers are equal and have identical efficiencies at each machine center. Many variations in labor efficiency patterns may be depicted in the heterogeneous systems, where the laborers do not have equal and identical efficiencies at each machine center. For the experiments described in this paper all DRC systems were homogeneous in nature only.

Labor Blocking. One of the phenomena that occurs in DRC systems is "labor blocking". This occurs where there are idle labors and there are jobs left in one or more queues. This can occur if there are empty machine centers and the only jobs waiting are in queues behind busy machine centers. If no jockeying between queues is allowed, we have both waiting jobs and idle workers. Such a situation can occur in reality where each machine center performs some special task that cannot be performed by the others. For example, a repair garage may have only one rig to realign the front-end of cars, one paint spray booth, etc.

The Model. The basic model described in this paper is a variation of GERT (Graphical Evaluation and Review Technique). Developed by Pritsker (18,19), GERT, which is similar to PERT, is a procedure for modeling stochastic decision networks. Since its inception, GERT has evolved through GERT II and GERTS III (a general purpose program written in GASP) for simulating stochastic networks. This was followed by GERTS III C, GERTS III Q and GERTS III R.

Basic GERTS. The general features of the basic GERTS simulation models include:

a) Network branches - characterized by the probability of being selected, the time required to complete the activity represented by the branch (it may have any one of several probability distributions), and the efficiency of each resource required to perform that activity (optional).

b) Nodes - characterized by number of "releases," before the node is realized or reached for the first time and after the first time, which activities must be completed for the node (event) to be achieved (since some branches have a probability of being selected, not all branches incident upon a node need be required), method of scheduling the activities emanating from the node, and the statistics to be collected (if any) at the node.

For network modeling purposes, each node may be classified in one or more of 10 categories. In this work only the following were used. (See [17] for a description of all types).

SOURCE Nodes: Nodes which initiate activities at the origin time of the project.

SINK Nodes: Nodes which may be the terminal node of the network. Normally SINK nodes only receive flow, but it is also possible to have activities leaving a SINK node if so desired.

MARK Node: This node is used as a time frame reference point for an item being processed. The point in time at which an entity passes a MARK node is recorded as an attribute to that entity. MARK statistics are collected at INTERVAL STATISTICS nodes and constitute the time spent in passing between the MARK node and the INTERVAL STATISTICS nodes.

QUEUE Nodes: QUEUE nodes are those which provide a storage capacity for items in progress. Items are automatically held at a QUEUE node until a service activity is performed on that item. Statistics are automatically maintained on QUEUE nodes.

STATISTICS Nodes: STATISTICS nodes are those at which statistical quantities are collected. There are five basic statistics which can be collected. Any node except START, QUEUE, or MARK node is a candidate for a STATISTICS node.

Figure 1 shows the node symbolism for normal GERT nodes and Figure 2 gives the notation for QUEUE nodes.

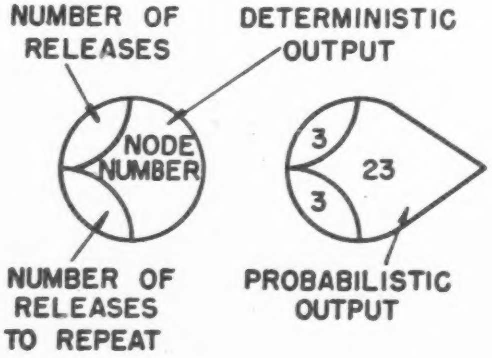

NUMBER OF RELEASES
DETERMINISTIC OUTPUT
NODE NUMBER
3
23
3
NUMBER OF RELEASES TO REPEAT
PROBABILISTIC OUTPUT

FIGURE 1

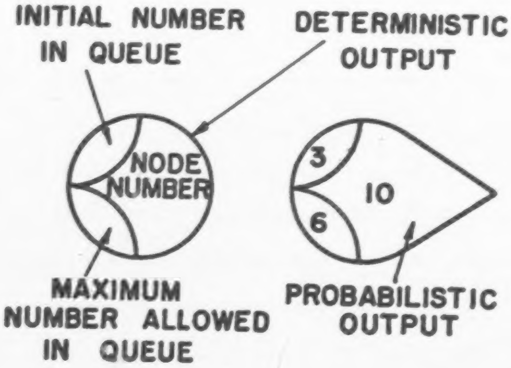

INITIAL NUMBER IN QUEUE
DETERMINISTIC OUTPUT
NODE NUMBER
3
10
6
MAXIMUM NUMBER ALLOWED IN QUEUE
PROBABILISTIC OUTPUT

FIGURE 2

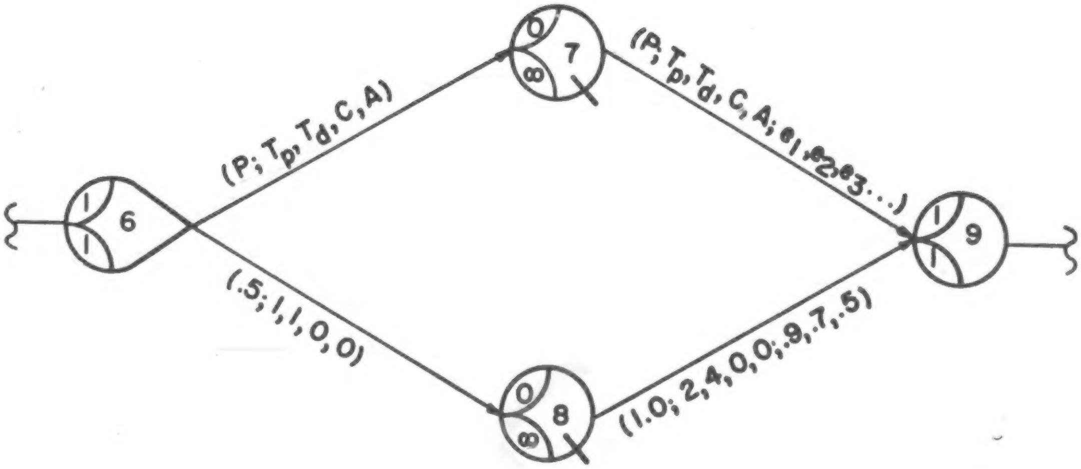


FIGURE 3

Figure 3 further illustrates the node symbolism and also displays the method for specifying the arc parameters. Figure 3 shows a variety of node and arc types, for which a brief explanation follows. Node 6 is a node which has probabilistic output, thus the probability of selection (P) is less than one on arcs (6,7) and (6,8). The parameter set format is defined on arc (6,7) and a typical numerical example is given on arc (6,8). Here the probability of selection is .5; the time of traversal is given by the first parameter set and first distribution type, and there is no counter or activity number assigned. Both nodes 7 and 8 are queue nodes with initial queue length of zero and infinite queue capacity. Note that the output arcs of these nodes have additional parameters, e_1, e_2, ..., e_n; these parameters give the efficiency of each resource in performing the activity. The parameters of arc (8,9) show that the first laborer has an efficiency of .9; the second .7; and the third .5 when performing this activity. Further explanation will be given in the context of the GERT III QR network examples in the next section.

The GERTS III QR simulation model was constructed using GERTS III Q as a framework, and integrating the concepts of GERTS III R with some modification. GERTS III R was designed for the study of activity networks with limited resources; i.e., most arcs represent activities which are resource constrained. So in the processing of the network, GERTS III R assumes that the resource constraints act upon every arc. In most queueing networks, the only arcs which might require resources are the output arcs of Q-nodes, which represent the service activity associated with the queue. Most often the other arcs of a queueing network merely represent flow paths for the items flowing through the system. Thus, considering every arc to be labor constrained could lead to much useless monitoring; therefore, GERTS III QR considers only output arcs and nodes in the normal GERTS III manner.

Another important change in the basic concepts of GERTS III R is the incorporation of the facility to deal with heterogeneous labor forces. In GERTS III R a number of resource types is specified (in the present version as many as three, although a special application to be reported later uses ten resource types) and an activity requires a fixed number of units of each type. The resource types are not interchangeable and each type is equally efficient on all activities. In GERTS III QR resources are interchangeable; that is, if an activity requires units of some resource any type can be used. Thus the type merely identifies the resource. Further, the time required to complete an activity is a function of the resource type (laborer) which performs the activity and the identity of activity (service facility). Hence, complex patterns of labor and machine efficiency can be studied.

Some modification of the basic concepts of GERTS III Q was also required. In DRC queueing systems, items may be detained in the queue not only because the service facility is busy but also because no labor is available, so that a service facility may be idle while items are in queue. Thus the basic queueing concepts had to be modified to deal with this dichotomy which exists in the DRC situation.

Hence, it is inaccurate to say that the GERTS III QR model is formed by mere superposition of GERTS III R upon GERTS III Q. Although the GERTS III QR simulator was written for DRC systems it will simulate ordinary machine limited systems with little or no loss in computational efficiency compared with GERTS III Q. However, for networks where practically all arcs are resource constrained and no queues are involved the GERTS III R simulator is more effective. GERTS III QR represents an integration of all the standard features of GERTS III R and GERTS III Q.

A standard feature of GERTS III QR is the QUEUE node. The QUEUE node is one which provides a storage capability for on-going items. The concepts of first and secondary releases are not appropriate for a QUEUE node, and so the QUEUE node is characterized by: (1) the number of items initially in the queue, and (2) the maximum number of items allowed in the queue. Other parameters associated with a QUEUE node are the order of processing and the node to which an item would balk if it arrives when a queue is full.

In the version used in this study, GERTS III QR was further modified to include priority rules other than the standard rules, LIFO and FCFS. To implement the SOT rule additional attributes had to be added to the queue list since job processing times were assigned to the job upon entry into the system. Therefore, a job selection sub-routine was also now required.

Implementation of the RANDOM priority rule required only minor modification of the SOT subroutines. The initial randomly assigned job times were used as the random variable for job priority selection using the SOT procedure, however, a different job processing time was then assigned when the job underwent actual processing.

For STATISTICS nodes, GERTS III QR obtains estimates of the mean, standard deviation, minimum, maximum and a histogram associated with the time a node is realized. Five types of time statistics are possible:

F. The time of first realization of a node;

A. The time of all realizations of a node;

B. The time between realizations of a node;

I. The time interval required to go between two nodes in the network; and

D. The time delay from first activity completion on the node until the node is realized.

The nodes on which statistics are to be collected and the type of statistics desired are part of the description given to a node by the input to GERTS III QR.

A distribution type and parameters are assigned to an arc through the specification of a parameter set number and a distribution type. Each parameter set defines parameters from which the mean, variance, maximum value and minimum value for each of the above distribution types can be computed. This description is part of the data input to the GERTS III QR program.

A powerful device when using the GERTS III QR program to analyze complex activity networks is the ability to modify the network while an activity is in progress. Specification of an activity number allows network modifications based on the completion of specified activities within the model. An activity may or may not be numbered. However, only those activities which are numbered are candidates for network modification.

Standard output of the GERTS III QR programs consists of the following:

1. An echo check of the input data, consisting of:
    A. Node characteristics
    B. Branch characteristics
    C. Listing of the SOURCE, SINK, and STATISTICS nodes
    D. Network modifications
2. Statistical summaries consisting of:
    A. The probability of node realization during the simulation period. The mean, standard deviation, number of observations, maximum, and minimum time units to realize a STATISTIC node during the simulation.
    B. All of the above statistics for counter types.
    C. Mean, standard deviation, minimum and maximum of the queue length, waiting time, busy time of processors (service

activities), and balkers per unit time, for all QUEUE nodes.

D. Histograms of the time to realization for each STATISTICS node, and the queue lengths for each QUEUE node. Histograms reflect the underlying probability distributions associated with All, Interval, Delay, Between, and First realization statistics.

It should be noted that the above output quantities are common to all GERTS simulation programs, and the interpretation of each statistic has been previously discussed in prior publications (1,2,20). Hence, in the examples which follow, only the output statistics unique to GERTS III QR will be discussed.

A GERTS III QR Network Model for Multi-Queue, Multi-Channel, Single-Phase, DRC Queueing System: An Example

In this section an application of the GERTS III QR model will be illustrated; the systems modeled are multi-queue, multi-channel, single phase systems. Each service facility contains one machine and has its own queue. Balking or switching between queues and reneging are not allowed. The arrival process is Poisson with a mean arrival rate of $\lambda = 1.0$. Each job is routed to a specific service channel with equal probabilities, $1/m$, upon arrival. Every job requires only a single processing operation at the service facility to which it is routed. The mean processing times are assumed to be identical and exponentially distributed. The number of laborers, n, is less than the number of machine centers, m. Thus the system is constrained by both labor and machines. Labor is allocated to available jobs by assigning the most efficient laborer to the machine center with the job having the highest priority per the selected priority rule. In this example the labor force is assumed to be homogeneous and all laborers' efficiencies are equal to 1.0 (100%).

The Problem Statement

The problem can be stated as follows. Given the system described above, determine the proper job selection rule (machine loading) and its related labor assignment rule so as to optimize some measure of system performance. The key measures used in this study are job waiting time characteristics.

The machine loading and labor assignment rules to be evaluated are: (1) select jobs from the queue per the first-come-first-serve (FCFS) rule and assign idle laborers accordingly, (2) select the job from the queue with the expected shortest-operation-time (SOT) and (3) select jobs at random (RANDOM). A schematic representation of the experimental systems is given in Figure 4.

This illustration, representing a typical labor limited (DRC) system, is merely one example of the application of GERTS III QR. Certainly many other system descriptions could

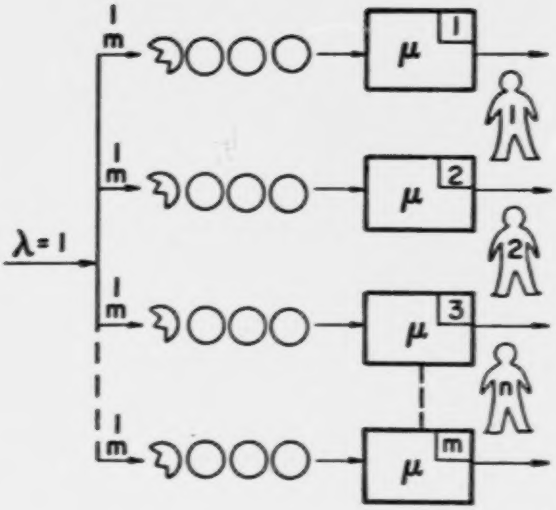


FIGURE 4

have been chosen as the technique is very general in structure. However, this particular system was chosen for three major reasons: (1) there are many actual situations which are closely approximated by this model; (2) the system is complex enough to illustrate the power of GERTS III QR and yet simple enough for reasonably compact discussion; and (3) a study was conducted on systems of this type using Nelson's (9) simulator; so a comparison of simulator effectiveness can be made, as well as some verification of the model.

Upon further examination of Figure 4 one may see why the network approach was applied to systems of this type: the system schematic itself suggests a network model. Thus the network description is conceptually appealing and investigation in this area seems natural. The GERTS III QR network model for a 3 service facility and 2 laborer system is depicted in Figure 5.

Figure 5 depicts a GERTS III QR network model for a system with three machines and two laborers. The accompanying set of system and arc parameters for this example are given in Table I. The data in this table and the figure specify that this system has three machines and two laborers. For convenience this is referred to as a 3-2 system; the first number being the number of machines, m, and the latter being the number of laborers, n.

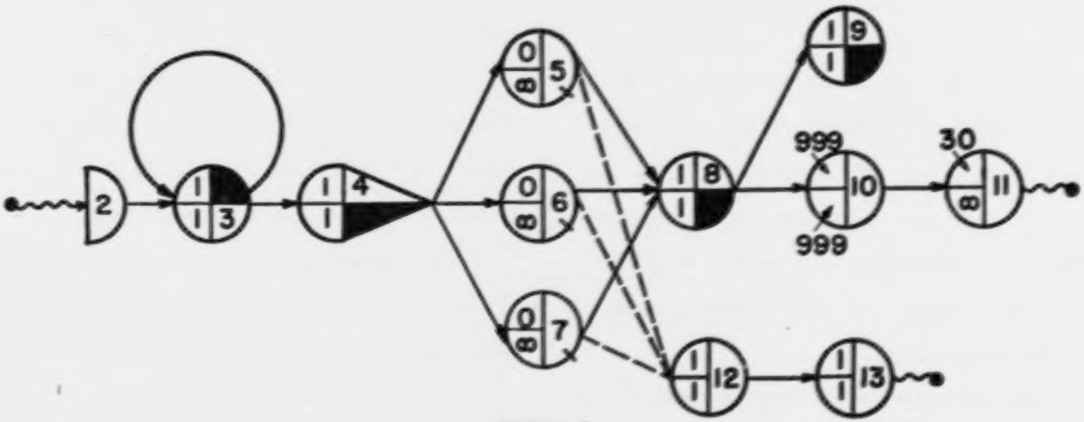


FIGURE 5

TABLE I

System and Arc Parameter for Example 1

System Parameters

Total number of nodes = 12
Number of sink nodes = 2
Number of nodes to realize the network is 1

Number of resources (laborers) = 2
Number of source nodes = 1
Statistics collected on 5 nodes

Arc Parameters

Parameter Sets

(Probability of realization; distribution of traversal time; labor efficiency)

Arcs	
(2,3)	(1.0; constant time of zero; no resources)
(3,3)	(1.0; exponentially dist. with mean = 1.0; no resources)
(3,4)	(1.0; constant time of zero; no resources)
(4,5)	(1/3; constant time of zero; no resources)
(4,6)	(1/3; constant time of zero; no resources)
(4,7)	(1/3; constant time of zero; no resources)
(5,8)	(1.0; exponentially dist. with mean = 1.8; $e_1 = 1.0$, $e_2 = 1.0$)
(6,8)	(1.0; exponentially dist. with mean = 1.8; $e_1 = 1.0$, $e_2 = 1.0$)
(7,8)	(1.0; exponentially dist. with mean = 1.8; $e_1 = 1.0$, $e_2 = 1.0$)
(8,9)	(1.0; constant time of zero; no resources)
(8,10)	(1.0; constant time of zero; no resources)
(10,11)	(1.0; constant time of zero; no resources)
(12,13)	(1.0; constant time of zero; no resources)

Experimental Design

The two basic building blocks in these experiments are the two machine, one laborer system and the three machine, two laborer systems. The 2-1/FCFS System can be solved analytically since it is equivalent to the 1-1/FCFS system. In fact, it was one of the methods used to validate the simulation model. The 2-1/SOT system was also solved analytically and used as a check of the model. (The analytical solution for this system is not readily found in the literature. An equivalent form was found in Saaty (21) and is attributed to the work by Phipps.)

The 3-2 system is the smallest truly labor limited system. All subsequent systems were designed to be multiples of these two basic systems.

For each system (2-1, 3-2, 4-2, etc.) a total of 26,000 jobs were simulated with the first thousand jobs discarded to initialize the system. In order to compare systems, at equivalent levels of utilization, the arrival rate was held constant and the mean service rate was adjusted to compensate for the varying number of labors. System utilizations of 0.75, 0.90 and 0.95 were selected for study.

As proposed by Nelson (14), system utilization, e.g. average labor utilization, may be estimated by equation 1.1 (in a simplified form):

$$1.1 \quad \hat{\rho}_{\ell} = \lambda/n\mu$$

Measures of System Performance

System performance is measured in terms of the "normalized" system mean waiting time, $\bar{w}/\bar{s}$, i.e., the actual mean waiting time, $\bar{w}$, measured in units of one mean service time, $\bar{s}$, as obtained from the simulations. As the experimental labor and machine limited systems currently defy analytical solution, the normalized mean waiting time values were obtained from the results of the simulations.

The Experimental Results

In Table II, the results using the FCFS, SOT and RANDOM machine loading and labor assignment rules are summarized. The main control variables were system configuration and labor utilization.

From the Table, it is evident that the SOT rule is more efficient in terms of normalized mean waiting time and mean flow time, than the FCFS and RANDOM rules. These results are intuitive as is the result that SOT yields a higher variance of flow time than FCFS. This rule quickly processes the short jobs and thus reduces both wait and flow times. However, those jobs with long estimated processing job times usually have a long wait, hence the higher variance. In comparing the variances for SOT and RANDOM, SOT has lower variance for systems of size 4-3 or greater. With respect to labor blocking, RANDOM had the greatest amount followed by SOT with FCFS having the lowest amount.

TABLE II

Waiting, Flow Time and Labor Blocking Characteristics of the Selected Priority and Assignment Rules for the Experimental Systems

			FCFS				SOT				RANDOM			
Run Number	SYST Configuration m-o	Estimated Utilization $\rho\ell$	Normalized Mean Waiting Time w/s	System Mean Flow Time MFT	Variance System Flow Time σ^2_F	Labor Blocking Hours B	Normalized Mean Waiting Time w/s	System Mean Flow Time MFT	Variance System Flow Time σ^2_F	Labor Blocking Hours B	Normalized Mean Waiting Time w/s	System Mean Flow Time MFT	Variance System Flow Time σ^2_F	Labor Blocking Hours B
1	2-1	.75	2.8170	2.9434	8.632	0	1.5839	1.9342	11.17	0	2.4627	2.5876	11.8626	0
2	2-1	.90	8.9722	9.0322	108.36	0	3.2375	3.8201	156.44	0	6.9245	7.1327	106.6022	0
3	3-2	.75	1.6540	4.0363	12.785	3,062	1.1074	3.1718	17.95	7,207	1.6477	3.9704	22.1177	9,646.44
4	3-2	.90	3.8167	8.7767	54.56	2,383	2.1676	5.7146	186.11	5,334	4.7777	10.3923	378.6786	6,573.49
5	3-2	.95	7.5662	16.3209	196.36	1,436	3.2044	8.0028	744.[illegible]	3,157	8.9884	16.9937	1407.6952	3,523.96
6	4-2	.75	1.7164	4.1791	15.72	2,481	0.9758	2.9745	15.25	4,995	1.3918	3.5762	19.5186	6,795.01
7	4-2	.90	4.2369	9.5089	69.50	1,427	2.0108	5.0327	160.57	3,797	3.3636	7.8485	147.0993	4,681
8	4-2	.95	8.2962	17.7143	183.60	678	3.0524	7.7142	662.41	2,231	7.4744	16.1012	1034.8303	2,789.02
9	4-3	.75	1.5098	5.6777	28.33	15,247	1.0530	4.6351	37.06	12,704	1.5544	5.7480	56.7077	41,408.55
10	4-3	.90	3.6994	12.7286	124.35	8,233	1.9670	8.0104	282.15	20,018	3.8068	12.9790	741.0416	23,157.11
11	4-3	.95	11.6595	36.1677	1263.81	3,178	2.8624	11.0301	1103.90	11,384	7.2490	23.5234	4219.2482	12,419.25
12	6-3	.75	1.1407	4.9636	19.23	7,612	0.7892	4.6351	37.06	12,704	1.0473	4.6030	30.1960	22,447.76
13	6-3	.90	2.9908	10.8921	74.81	4,102	1.5379	6.8270	165.25	12,147	2.2549	8.7786	192.2134	11,516.07
14	6-3	.95	4.1985	16.9873	124.41	2,875	2.3111	9.4592	647.41	7,658	4.7083	16.2817	1371.9761	8,669.71
15	6-4	.75	1.2064	6.6849	36.78	24,292	.8753	5.6677	44.30	63,039	1.1162	6.3464	63.6137	86,828.87
16	6-4	.90	3.0821	14.7855	171.95	12,422	1.5557	9.2272	259.99	18,799	2.0622	13.9128	760.7598	40,689.83
17	6-4	.95	5.2960	21.8385	356.26	6,805	2.2636	12.4386	940.82	20,270	4.6681	21.9476	3342.8011	22,382.82
18	8-4	.75	0.8992	5.7761	28.24	18,048	0.6816	5.0651	31.79	39,161	0.8583	5.5665	40.2196	53,931.64
19	8-4	.90	2.3039	12.0246	97.44	9,569	1.3075	8.3326	181.63	25,878	2.1318	11.3076	537.8106	29,231.72
20	8-4	.95	3.8669	18.5936	213.04	5,282	1.9166	11.3825	680.42	14,717	3.5549	17.2971	1733.1064	17,197.83
21	8-6	.75	1.3788	10.7384	111.32	110,604	0.9231	8.6866	106.98	377,308	1.1969	9.98	195.2367	564,578.13
22	8-6	.90	2.6998	20.0477	289.74	42,726	1.5645	13.8893	502.80	190,453	3.0855	22.1091	2375.6958	133,778.41
23	8-6	.95	5.0651	34.6755	702.96	13,230	2.1852	18.1993	1822.31	54,634	4.2860	30.1082	7854.2246	78,452.48
24	9-6	.75	1.0823	9.1184	73.61	85,517	.7892	8.0839	81.05	270,232	0.9609	8.8256	129.5426	363,357.71
25	9-6	.90	2.6499	19.8192	286.41	30,963	1.3480	12.7188	353.11	139,110	2.0591	16.5388	1130.5102	197,385.25
26	9-6	.95	3.6460	26.5558	435.17	17,838	1.8917	16.5241	1209.32	54,008	4.1337	29.3176	11148.0962	67,011.67

A somewhat surprising result is that the amount of labor blocking is higher for the SOT rule than the FCFS rule. The authors have no readily intuitive explanation for this phenomenon, and it is a subject for further research. Another effect is that for a given system with all rules, labor blocking drops as system utilization increases. Again, this is an intuitive result since labor blocking is a phenomena associated with idle laborers and this (idleness) decreases with higher systems utilization.

Labor Effectiveness

To compare selected experimental results with computed analytical results let's refer to Figure 6.

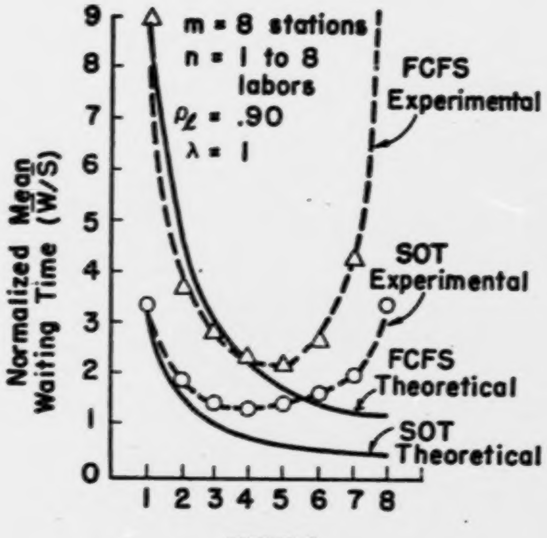


FIGURE 6

In Figure 6, the number of service centers, m, is fixed at 8, $\hat{\rho}_\ell$ = .90, and the number of laborers, n, varies from 1 to 8. The experimental change in $\bar{w}/s$ resulting from a change in n is depicted by the u-shape curves. In addition to the experimentally obtained u-shape curves, curves for the "theoretical normalized mean waiting time" using FCFS and SOT are shown. The data points for these curves are given by the equation:

$$1.2 \quad (\bar{W}/\bar{S})_n = W_1\mu_1/n$$

where

$(\bar{W}/\bar{S})_n$ = the "theoretical" normalized mean waiting time for a system with n laborers.

$\bar{W}_1$ = the mean waiting time for a system with one laborer

μ_1 = the mean service rate when the system has only one laborer.

n = the number of laborers.

The "theoretical" values given by this formula assume that each laborer added to the system is equally as effective as the first laborer; i.e., this formula neglects the effects of both labor blocking and "flexibility" in the system. Here "flexibility" is measured by the number of machine centers to which an idle laborer may be assigned. By comparing this theoretical curve with the simulated values of normalized mean waiting time, one may get some conception of the effects of blocking and "flexibility" on the system performance.

If additional laborers were as effective as the first, the normalized mean waiting time would decrease as shown by the theoretical curve. Since the effect of labor blocking is present, one would expect that as the number of laborers increases, their effectiveness would decrease, i.e. labor blocking increases because there are fewer machine centers to which an idle laborer can be assigned. A single laborer, in fact, can greatly affect system performance. If one is very busy then two things happen: (1) he is not often available to help cover the remaining work space and (2) jobs in this queue are blocked from the other laborers. A true measure of "labor effectiveness" remains to be formulated and appears to be an interesting area for future research. Nevertheless, we can see from Figure 6, that the SOT rule consistently performs the best with respect to $\bar{W}/\bar{S}$.

In Table II are shown some comparisons between multiples of a basic system at the same level of utilization. (4-2, 6-3, 8-4 and 3-2, 6-4, 9-6). As can be seen, the normalized mean waiting time decreases as the size of the multiple increases. This is due to the fact that, for example, an 8-4 system is more efficient than two 4-2 systems since laborers can cross over to any of the 8 machines in the first, while constrained to stay within each set of four machines in the second system. The same effect and analysis applies to mean flow time. Among these three rules, again SOT gives the best results.

Computer Experience

The simulations were run on a CDC 6600 computer at the University of Texas at Austin. A total of 26,000 jobs were simulated in each run. The first thousand jobs were not used in calculating the statistics but were required to initialize the system. The simulation run times ranged from 142 seconds for a 2-1 system to a maximum of 175 seconds for a 9-6 system. This includes 13.5 seconds for compile time. The run times for the FCFS and RANDOM systems were generally 20% higher than for the SOT rule. The core storage requirements were 40,200 words. For identical small systems the GERTS III QR simulator requires less than half the computation time required by the prior SIMSCRIPT simulator (9,14). For larger systems the savings is even more significant.

Conclusions

The results of this study show that for labor and machine limited queueing systems the GERTS III QR model is not only a feasible model but that it is a very efficient model. With respect to the example problem, the Shortest Operating Time rule is found to be superior to the First-Come-First-Served and RANDOM rules. This is also the case with purely machine limited systems. However, in the DRC systems a phenomenon exists in the form of labor blocking. This causes a change in normalized mean waiting time from the theoretical values.

The GERTS III QR simulator represents a powerful tool to simulate labor limited job shops and to evaluate various machine loading and labor assignment rules.

While the discussion in this paper centered upon parallel channel, single phase, homogeneous DRC experimental systems, GERTS III QR has been extended and used in a real world application. At a major air force maintenance base this model has been used to plan the work flow of aircraft engines through overhaul. The network consisted of three major maintenance lines in parallel with each line being a network of service centers in parallel and series, requiring up to ten resource classifications. Each line consists of at least 40 nodes and the entire network includes 132 service centers.

The preliminary runs were used to identify both bottlenecks and greatly underutilized service centers. This facilitated a shift of resources throughout the network in order to achieve a more balanced workload.

The experiments reported in this paper are being duplicated for heterogeneous DRC systems. These latter investigations have led to the evoluation of some interesting and practical labor assignment rules and they will be the subject of a future paper.

In summary, GERTS III QR is a major improvement over the more traditional simulation packages for ease of structuring systems and in savings of computer time.

* * *

We wish to acknowledge the work done by Hogg (8) and Hoge (7) in developing the statistics on FCFS and SOT respectively.

REFERENCES

1. Burgess, R. R. and A. A. B. Pritsker, "Definitions and Listings for GERTS III, GERTS III Q, GERTS III C, and GERTS III R," NASA/ERC Contract NAS-12-2113, Virginia Polytechnic Institute, (May, 1970).
2. Burgess, R. R. and A. A. B. Pritsker, "The GERTS Simulations Programs: GERTS III, GERTS III Q, GERTS III C, and GERTS III R," NASA/ERC Contract NAS-12-2113, Virginia Polytechnic Institute, (May, 1970).
3. Conway, R. W., "An Experimental Investigation of Priority Assignment in a Job Shop," Memorandum RM-3789-PR, The Rand Corporation, Santa Monica, California, February, 1964.
4. Harris, Roy D., "An Empirical Investigation and Model Proposal of a Jobshop-Like Queueing System," Western Management Science Institute, Working Paper No. 84, University of California, Los Angeles, July, 1965.
5. Harris, Roy D., "A Manpower Limited Service Center Model," Working Paper No. 68-14, Graduate School of Business, University of Texas at Austin, November 1967.
6. Harris, Roy D. and Michael J. Maggard, "Empirical, Analytical and Simulation Studies of Labor Limited Production Systems," Working Paper No. 70-16, The Graduate School of Business, The University of Texas at Austin, November, 1969.
7. Hoge, James, "A GERT Simulation of Labor Limited Queueing Systems Using the Shortest Operating Time Priority Rule," Unpublished Professional Report, Graduate School of Business, The University of Texas at Austin, July, 1972.

8. Hogg, Gary L., "An Analysis of Labor Limited Queueing Systems with a GERT Simulation," Unpublished Ph.D. dissertation, University of Texas at Austin, 1971. Available from University Microfilms, Ann Arbor, Michigan.

9. Maggard, Michael J., "An Evaluation of Labor and Machine Limited, Parallel Queueing Systems," Unpublished Ph.D. Dissertation, University of California, Los Angeles, 1968. Available from University Microfilms, Ann Arbor, Michigan.

10. Maggard, Michael J., "An Evaluation of Labor and Machine Limited, Parallel Queueing Systems: Part 1: Design Implementations," Working Paper 69-63, Graduate School of Business, The University of Texas at Austin, February, 1969.

11. Maggard, Michael J., "An Evaluation of Labor and Machine Limited, Parallel Queueing Systems: Part 2: Economic Models," Working Paper 69-68, Graduate School of Business, The University of Texas at Austin, April, 1969.

12. Nanot, Yves R., "An Experimental Investigation and Comparative Evaluation of Priority Disciplines in Job Shop-Like Queueing Networks," Management Sciences Research Project, Research Report No. 87, University of California, Los Angeles, December, 1963.

13. Nelson, Rosser T., "An Empirical Study of Arrival, Service Time, and Waiting Time Distributions of a Job Shop Production Process," Management Sciences Research Project, Research Report No. 60, University of California, Los Angeles, June, 1959.

14. Nelson, Rosser T., "Labor and Machine Limited Production Systems," Management Science, Vol. 13, No. 9, pp. 648-671, May, 1967.

15. Nelson, Rosser T., "Dual Resource Constrained Series Service Systems," Operations Research, Vol. 16, No. 2, pp. 324-341, March-April, 1968.

16. Nelson, Rosser T., "A Research Methodology for Studying Complex Service Systems," AIIE Transactions, Vol. I, No. 2, pp. 97-105, June, 1969.

17. Phillips, Don T., Gary L. Hogg, and Michael J. Maggard, "GERTS III QR: A GERTS Simulator for Labor Limited Queueing Systems," Research Memorandum No. 72-2, School of Industrial Engineering, Purdue University, Lafayette, Indiana, April, 1972.

18. Pritsker, A. A. B. and W. W. Happ, "GERT: Graphical Evaluation and Review Technique; Part I - Development," The Journal of Industrial Engineering, Vol. 17, No. 5, May, 1966.

19. Pritsker, A. A. B. and G. E. Whitehouse, "GERT: Graphical Evaluation and Review Technique; Part II - Probabilistic and Industrial Engineering Applications," Journal of Industrial Engineering, Vol. 17, No. 6, June, 1966, pp. 45-50.

20. Pritsker, A. A. B. and Don T. Phillips, "GERT Network Analysis of Complex Queueing Systems," Research Memorandum No. 72-1, School of Industrial Engineering, Purdue University, Lafayette, Indiana, January, 1972.

21. Saaty, Thomas L., Elements of Queueing Theory with Applications, McGraw-Hill Book Co., Inc., New York, 1961.

22. Takacs, Lajor, "Two Queues Attended by a Single Server," Operations Research, Vol. 16, No. 3, pp. 639-650, May-June, 1968.

EVALUATING JOB SHOP SIMULATION RESULTS

Carter L. Franklin, II

Assistant Professor of Management
Wharton School
University of Pennsylvania

Abstract

The problem of predicting the effectiveness of simulation results as applied to job shop production is discussed. A measure of effectiveness is developed which allows both absolute and relative evaluation of different scheduling techniques and the conditions for profit maximization under a customer service constraint are presented. The development is based upon a model of the shop process which assumes random scheduling behavior and which serves as the reference point for evaluating schedules.

I. Introduction

The problem of scheduling production is the central element of the production control problem for the job shop. Production performance is almost entirely determined by the quality of schedules produced. Research and applications studies on the problem have, almost entirely, been either simulations or simulation-based. The use of simulation is mandated by the complexity of the problem and the lack of an algorithm for finding solutions. The importance of the scheduling problem is widely recognized by members of both the academic and the industrial community.

The solution to the job shop scheduling problem has its ultimate value in application. The scheduling rule or procedure, which is the solution, may be used in either a manual or computer based system to effect maximal production performance. The objectives served by the production control function in the job shop, through the schedule, are: (1) profit; (2) customer service; (3) reduced investment in work in process inventories; and (4) improved utilization of capacity. Conway, Maxwell, and Miller [1] discuss these objectives, pointing out that direct costs are irrelevant to the scheduling decision and that only indirect costs form the proper basis for evaluating schedules.

The search for the best scheduling rule has gone on for many years. The work that has been done may be placed in two broad categories: "Academic Research" and "Industrial Applications". Much of the progress made in one cateogry is of little value in the other. Results and findings are not seen as transferrable and, as a result, are not transferred. The academic studies are devoted to problems too highly abstracted and bearing little relation to those found in industry. Industrial applications are too specialized, conforming to the special and possibly unique needs of the company which developed the application. In addition, results and details of system structure are often viewed as proprietary and are not publicized. Reports of effectiveness, when made available, are often without meaning when taken out of context. Thus, when one company can claim to have reduced lateness by a half or a third through scheduling, another company has no guarantee that the same, or a similar, system will produce the same result for them.

II. The Problem of Measurement

A major source of difficulty in assessing the transferability of results between two industrial applications or between academia and the industrial community is the lack of a good and generally acceptable measure for evaluating schedule quality or performance. The problem is compounded by the fact that academic studies are typically couched in purely technical terms (flow time is the most common measure) with no reference to economic criteria, and that industrial applications are described primarily in terms of the change in economic performance achieved. A means of constructing a true cost effectiveness measure for changes in schedule quality must be available if scheduling studies and applications are to effect general changes in manufacturing performance. The cost of a scheduling study or application cannot be determined in the absence of information about costs of computation, programming and manpower. The effectiveness of a scheduling rule or system, in application, can be found both in absolute terms and in terms of the improvement over current procedures.

The requirements for the measure of effectiveness, or benefit, are neither simple nor easily satisfied. Although many requirements of varying importance may be stated, there are four which are of primary importance:

(1) Schedule performance must be measurable in both economic and technical terms. In addition, the conversion from an economic

to a technical measure, and vice versa, must be easily accomplished.

(2) The effectiveness measure must be general and widely applicable. It must not be tied to unique or special economic or technical characteristics.

(3) The measure must be capable of being easily understood by managers, both in terms of technical and economic performance.

(4) The measure must be applicable to both actual and simulated performance. That is, it must not require information not readily available in a job shop.

The last of these requirements can be of particular importance, especially since the motivation to undertake a scheduling study or application most often arises from an evaluation of the effectiveness of procedures currently in use.

Every scheduling procedure or production control system, whether manual or computer based, presumably exhibits some of the characteristics of intelligent, goal seeking behavior. Under this premise, the performance under any such system should exceed that available from a completely random system; i.e., one in which scheduling decisions are made on a random basis. In the following sections, the measure of effectiveness will be developed and it will be applied to a random system so that a baseline for comparison will be available.

Job shop scheduling simulations, whether abstract or applied, do not lead to normative results in the sense that optimal schedules or scheduling rules can be specified. This, however, should not imply that a normative, profit maximizing, criteria should not be used as a basis of comparison for simulation results. Normative measures have not been either developed or applied in job shop research or application and progress toward the ultimate solution to the "job shop problem" has suffered as a result.

III. The Job Shop Process

Job Shop production is difficult to characterize because of the variety and complexity of products produced, both to customer order and to inventory. No measure of output, other than a generalized measure of products produced per unit time, is available. Inputs are similarly difficult to characterize since they may be unique to each product. (This is particularly true for materials.) A generalized model of the job shop process must, for these reasons, be aggregated and stated in terms of cost and resource flows. This does not present great difficulties since our evaluation of the job shop process is, primarily, couched in terms of indirect resources and their costs. Since it is the direct resources required by products and the processing requirements for individual products that gives the job shop problem its complexity, we can avoid much of this complexity and the problems it creates through aggregation.

The job shop process is amenable to analysis under the structure of Figure 1.

The job shop process, as represented by Figure 1, is that of taking the inputs machinery, labor and supervision (in the required proportions) as the productive resource "capacity". To capacity is added the input materials and capacity is transformed into the resource "work in process." The resource work in process in then transformed into units of completed products which are represented in revenue terms. This model of the job shop process is unique, perhaps, since only capacity and work in process are viewed as productive resources (which have productivities). The inputs machinery, labor, supervision, and materials are not individually productive. In this model, demand occurs at a rate (in products per unit time) and the average processing requirement for products is P (in units of time per product). The resources are M, the number of machines (or points of processing capability), and N the number of products in process. The productivities of the resources are p_f, the utilization of capacity, defined as

$$p_f = \frac{\lambda P}{M}$$

and p_d, the ratio of mean processing time to mean flow time (F) for products[1],

$$p_d = \frac{P}{F} .$$

Assuming that the job shop system is a queueing system and that steady state conditions obtain, then $N = \quad F$ and p_d may be written as

$$p_d = \frac{\lambda P}{N} .$$

At steady state, $Np_d = Mp_f$ and output, X (in products per unit time) may be written as

$$X = \frac{Np_d}{P} = \frac{Mp_f}{P} .$$

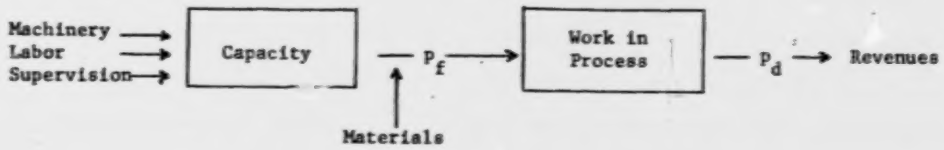


Figure 1

The Job Shop Process

---

[1] Technically, the productivities are p_f/P and p_d/P, but the denominators cancel in all cases and are ignored. Both p_f and p_d are between zero and one under steady conditions.

IV. Measuring Effectiveness

Control over the job shop process under this model may be effected by controlling M. Under the assumptions stated, control is exercised so that profit is maximized subject to a constraint on customer service, stated as a mean flow time requirement, $\bar{F}$. The point at which maximum profit occurs is that at which the ratio of the marginal productivity to marginal cost is equal across all resources.[1] To compute these ratios, knowledge of marginal cost must be available. These are simply the incremental costs of having additional units of work in process and additional units of capacity.[2] It should be noted that since both N and M are time averaged, the increment costed need not be integer and, in the case of M, a new machine need not be purchased if overtime is available. Letting C_d and C_f be the marginal costs of incremental work in process and capacity, respectively, profit is maximized when

$$\frac{p_d}{C_d} = \frac{p_f}{C_f} .$$

This may also be written as $MC_f = NC_d$. The optimal operating configuration is found in the following way. First $\bar{N}$ is calculated from $\bar{N} = \lambda\bar{F}$. The optimal value of M is then computed as $\bar{M} = \bar{N}C_d/C_f$. From $\bar{M}$ and $\bar{N}$, both p_f and p_d may be found.

These relations may be used to evaluate simulation results in a straightforward manner. Let us first assume that the simulation is conducted for potential application of a scheduling rule to a shop with known cost data. It should be noted in passing that, under a manual control system acting to maximize profits in the manner described above, it is extremely difficult to maintain the optimal operating configuration. It should not be assumed, for purposes of comparison, that a manual system achieves this state since the average effects of a set of individual control decisions, made serially, cannot always be controlled very well. Continuing the example, it is clear that the ratio p_d/p_f must be equal to C_d/C_f at the optimal operating point. The value of $\bar{p}_d$ (under the random assumption) is also known from $\bar{F}$. On a graph of p_f versus p_d, the ratio C_d/C_f may be drawn as a line through the zero point of the graph with positive slope. The intersection of this line with the line represented by $\bar{p}_d$ is the profit maximizing operating configuration. This graph is illustrated in Figure 2.

Simulation results, for a utilization of $\bar{p}_f$, should fall above the intersection of the $\bar{p}_d$

---

[1] See Samuelson [5], p. 60.

[2] The production system described by this model is stochastic and steady-state is assumed. Therefore, the best, and only, estimate of marginal quantities is their time average.

and $\bar{P}_f$ lines. If they do not, then the scheduling rules tested do not perform as well as the random rule. The simulation results for the load/capacity configuration represented by $\bar{P}_f$ may be plotted on the $\bar{P}_f$ line (the symbol "ⓧ" represents these on the graph). None of these represent optimal profit performance since none of them falls on the C_d/C_f line. Further simulations must be run until a result is found which falls C_d/C_f line at the highest value of P_f attainable. It is noteworthy that this result may not be the minimum average flow time result for the load/capacity ratio tested. It may also occur that the $\bar{P}_d$ line may not intersect the C_d/C_f line, particularly in cases when the value of C_d/C_f is small. When this occurs, there is no operating configuration or scheduling technique which will satisfy both the customer service constraint ($\bar{F}$) and the profit maximizing condition. In this case the management of the job shop has some (possibly) difficult choices to make.

Simulation studies conducted in the absence of economic information, obviously, cannot specify an optimal scheduling technique for any job shop with a customer service constraint. In the absence of economic information, performance of the several scheduling techniques tested must be represented by lines on the P_f, P_d graph and the determination of optimality must be deferred. The performance lines for a set of techniques will, in general, start in the upper left corner of the graph and slope downward to the right. These lines, particularly those representing a mixed scheduling technique (e.g., truncated SOT), may not necessarily be linear - or even continuous. A set of simulation results might be presented in the fashion illustrated in Figure 3.

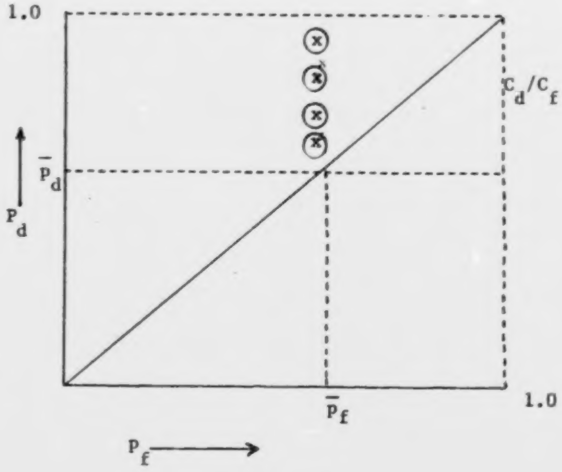


Figure 2

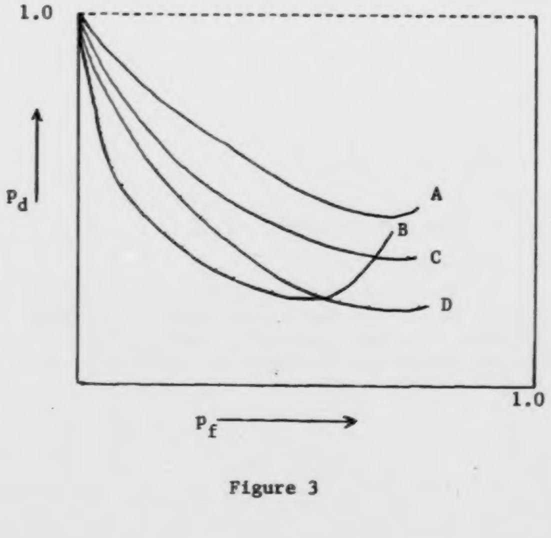


Figure 3

V. Data Acquisition

As might be expected, the primary data problems which arise in evaluating the effectiveness of simulation studies are not related to the simulation or its results. The only precautions which need be taken are those of insuring that steady state is achieved and that actual resultant values are found for P, λ, and N. The major difficulties arise in determining current shop performance (so that a point of comparison is obtained) and measuring the economic factors.

Since it cannot (or should not) be assumed that the performance of the control procedure in operation in the shop conforms to the structure developed under the assumptions of the random model, both technical (productivity) and economic measures should be made. There are several sources for this data, including the shop floor, the production control function and the financial statements of the firm. These sources may not prove to be consistent upon cross checking (which is highly recommended), creating problems which must be solved by closer examination of records or "by assumption."

Not the first, but probably the most significant problem that will be encountered is that of defining a job. A job, which may be several customer orders or a part of a single order, is a piece, or set of pieces, which follows a single route through the shop. The route need not be simple, but must be followed for the entire job. The balance of the technical data can be found by using one or more of the sources mentioned above. Information about the rate of demand or shipment (presuming steady state) can be obtained from the production controller and the accounting office. Information about mean processing time is available from sampling production control records and from supervisors on the shop floor (after determining the fraction of jobs processed in each work center). Data on capacity is available in the production control office and from observation on the shop floor. The number of jobs in process is most directly found by inspection and may be confirmed by the production controller. The average flow time may be found by sampling production control records and compared to the calculated value.

Economic information may be more difficult to obtain, depending upon the nature and availability of accounting and financial data. The unit marginal cost of work in process must, in the absence of specific information about the "next" jobs, be assumed to be equal to the average cost. Further, the average in-process job can be assumed to be half completed. It is reasonable in most cases to assume that all materials are committed when the job is started and that processing costs are directly proportional to processing time. Processing costs are the costs of direct labor and machine time that can be directly attributed to the job. It is important that these costs be direct, not allocated. The means for obtaining machine cost per unit of processing time will be discussed next.

The average unit cost of capacity is determined in the following way. First, the annual cost of supervision is determined and divided by the number of units of capacity. Next, the total annual depreciation and interest for productive machinery is calculated and divided by the number of units of capacity. Finally, the annual direct labor payroll is divided by the units of capacity. These are summed and further divided by productive days in the year to obtain the average daily cost of capacity. The average cost of capacity is not equal to the marginal cost of capacity. The machine cost component of capacity cost is zero at the margin if overtime or undertime production is available. The direct labor and supervisory components can likewise vary depending upon whether or not over or undertime is available or whether modifications to the size of the workforce must be made. Determination of the marginal cost of capacity is largely situation specific and, for the most part, depends upon the availability of overtime production.

The determination of work in process and capacity costs is not difficult in concept. In practice, some difficulties may be encountered because individual costs must be aggregated in a way not consistent with standard accounting practice. Hillier [3] discusses a similar cost finding problem and his paper is recommended to those who wish to pursue this problem further.

VI. Summary

The procedures for determining the effectiveness of job shop scheduling simulations meet all of the criteria stated in Section II above. The measures developed allow both relative and absolute comparisons for every case in which economic information is available. They increase both the effectiveness and desirability of job scheduling simulations because they allow the interpretation of simulation results in either a technical or an economic context. The measures lead directly to a statement of the conditions for profit maximization and bring all of us with an interest in this fascinating and complex problem a step closer to its ultimate solution.

REFERENCES

1. Conway, R. W., Maxwell, W. L. and Miller, L. W., Theory of Scheduling, Reading, Mass.: Addison-Wesley Publishing Co., 1967.

2. Franklin, C. L., "Current State of the Art in Job Shop Scheduling," Proceedings, Third Conference on Applications of Simulation, Los Angeles, California, December, 1969.

3. Hillier, F. S., "Cost Models for the Application of Priority Waiting-Line Theory to Industrial Problems," Journal of Industrial Engineering, Vol. XVI, No. 3, May-June, 1965. Also, in Buffa, E. S., ed., Readings in Production and Operations Management, New York: John Wiley & Sons, 1966.

4. Little, J. D. C., "A Proof for the Queueing Formula $L = \lambda W$," Operations Research, Vol. 9, No. 3, May, 1961.

5. Samuelson, P. A., Foundations of Economic Analysis, New York: Atheneum, 1967.

SIMULATION APPLIED TO A MANUFACTURING EXPANSION PROBLEM AREA

J. Douglas DeMaire

Olin Brass
Olin Corporation

Abstract

This study uses computer simulation to arrive at a specific answer to a critical problem area in a capital intensive manufacturing environment. A Casting Plant is being expanded due to growth in business. The logistics of the operation are such that an existing service facility, an overhead crane, will have added to its present workload, servicing an additional Casting Unit.

Management desired to know what magnitude of crane delays would result from increased task interference as a result of the increased workload on the crane. The answer to this question is crucial since certain types of crane delays generate a loss in casting capacity.

The simulation covers thirty-seven discrete types of tasks that the crane is required to do. These tasks are classified into three general types -

1) Fixed Time of Occurrence
2) Random Time of Occurrence
3) Random Time of Occurrence within some fixed distribution of known cycle times.

The simulation is designed to simulate operations for one-month, using one-minute time units.

INTRODUCTION

The manufacturing environment provides an extremely challenging area for applications of simulation. Many manufacturing problems become so involved and complex, that a tool like simulation is often a necessity to meaningfully capture the interrelationships of variables characteristic to the manufacturing environment. These complex interactions and the "real world" type problems of the manufacturing area are precisely the challenge. An additional facet that increases the challenge is dealing with manufacturing personnel in formulating and analyzing a simulator for a given situation. Manufacturing people, from necessity, must deal with a barrage of pressing daily problems and tasks. Therefore, careful project management is required to help insure the maximum utilization of the invaluable input the manufacturing people have to offer.

Through careful design and feedback control, a simulation study may be executed in such a way that the manufacturing people are "on board" the project from its design through its conclusion. The value of this direct involvement cannot be overstated; in addition to better parametic input to formulate the model, the results of the study have a much greater probability of actually becoming utilized input to the decision-making process. This fact results from these people's having a better understanding of, and therefore, more confidence and trust in the technique, as a result of their involvement throughout the development of the project.

The structure of this paper will basically follow the sequence of steps used in executing the project. Technical information is, for the most part, displayed in the appendixes, while the body of the report focuses more on the logistics of successfully executing the simulation study. Project structure is given below:

1. Problem Definition
2. Collection and Development of Model Parameters
3. Formulation of the Model - Present Situation
4. Benchmarking the Model
5. Addition of Proposed Equipment to Model
6. Conclusions and Results
7. Summary

A brief description of the manufacturing operation is given in APPENDIX C to aid in understanding the nature of the problem at hand.

PROBLEM DEFINITION

As a result of the business growth and predicted future growth, manufacturing capacity of the Casting Plant was projected to become insufficient to meet market demand. This projection signaled management that the casting operation had to be expanded.

Although the original design of the existing casting plant incorporated future expansion capabilities in casting equipment, it did not inherently provide for expansion of peripheral service equipment. One critical consideration in this area was concerned with overhead crane workload capacity. The present overhead crane was believed to be operating at a level near its capacity in the present layout.

Workload on the crane is critical to the casting plant operation. Certain types of crane delays occur as a result of crane task interference (more than one task requiring service simultaneously). These delays often cause an extension of the casting cycle and subsequently a loss in total casting capacity. With an estimated 20-25 per cent increase in crane workload projected as a result of the additional casting equipment being planned, a significant question began to crystalize. This question became a statement of the problem this simulation study would be called upon to solve.

How much casting capacity would be lost, without modification to the overhead crane operating system, as a result of the new casting equipment being brought on line? The answer to this question was the absolute objective of this study.

Several sub-questions required answers to help answer the major question and to provide other decision-making information to manufacturing management.

1. What magnitude of crane task interference would occur?
2. What new workload would result for the crane?
3. Would interference be great enough to justify extensive research to alleviate same?
4. Was it possible that no real problem existed?

After some analysis by manufacturing management, it became evident that the crane operating system was extremely complex and almost impossible to reliably analyze by conventional methods. Assistance in analyzing the situation was requested by manufacturing management, and it was specifically requested that a simulation model of the crane operating system be constructed.

The fact that manufacturing had specifically requested the simulation study proved to be extremely valuable throughout the study. This fact helped elicit good cooperation from the manufacturing operating personnel from the design phases to the conclusion of the project. This involvement of manufacturing personnel at all phases of the study served to inform them about the technique of simulation and, in turn, generated, in them, a trust in the method and its results, probably unattainable by any other method. In short, the results of the study were viewed as reliable inputs to the decision-making process by manufacturing.

Reiterating, the objective of the simulation became to measure the magnitude of the expected casting capacity loss due to the increased workload of the overhead crane. Given this measure, manufacturing would be in a position to make a more intelligient and well-informed decision as to what course of action should be taken.

DEVELOPMENT OF MODEL PARAMETERS

Manufacturing involvement in this phase of the project was critical. In essence, they would describe the system and their description would be abstracted into a simulation model. It is important to note that a model was not constructed and submitted for manufacturing approval; the model was developed with manufacturing an integral part of the development.

A series of meetings were held with operating and management personnel ranging from first line supervision through Director of Manufacturing. The initial meeting was somewhat unstructured. Generalities about the project were explained, and "rough" descriptions of system parameters were developed. At subsequent meetings, parameters were refined and expanded until all parametric information was developed.

Development of parameters consisted of accumulating information such as descriptions, detailed times for executing tasks, and descriptive information relative to distribution of occurrences of the various tasks.

Each crane task (16 discrete types) was detailed by sub-dividing it into sub-tasks. (See APPENDIX A) For example, one task might be broken down into 5 sub-tasks; each sub-task with a specified time for completion. Tasks were detailed in this manner in order that task priorities might be handled in the most realistic manner possible. Task breakdown and task priorities are discussed further in the FORMULATION section of this paper.

After several meetings with manufacturing, a sufficiently complete set of crane activities and their descriptions were compiled. At this point, a study of several months' past production and maintenance history was done, using the information previously developed as a basic structure for gathering the data. For example, given the various tasks, the study was used to develop occurrence frequencies for the tasks, based on history.

The results of the historical study were combined with manufacturing's best "operating feel" to yield the best estimate of the system parameters involved in this simulation. Specifically at this point, all crane tasks were identified and defined, frequency distributions for occurrences for each type task were complete, the nature of the occurrence distribution was known, and priorities had been assigned to the tasks.

FORMULATION OF THE MODEL - PRESENT SITUATION

Formulation of the model combined the parametric information that had been previously developed with the necessary logic to properly represent the interactions of the various components of the model.

A major part of the program logic required was knowledge about logical sequence of occurrence. That is, what events could logically occur at the same time and which events must occur at mutually exclusive times. Through cooperation with operating personnel, a matrix was developed that fixed which events could not happen simultaneously. (SEE APPENDIX B)

The other major part of logic development used in the model, dealt with generating tasks by the proper distributions. During this phase, it became clear that the crane tasks that had been defined could be classified into three types with respect to the nature of the occurrence distributions. These classifications were:

1. Totally random occurrences.
2. Totally random occurrences within some known but varying cycle time.
3. Fixed time occurrences.

The simulation program was developed or structured into three main sections that correspond to the three occurrence types.

A specific example of each occurrence type might be helpful.

The totally random events are exactly what the name implies. There is a known fixed number of occurrences of this type event, but the time of an occurrence is generated in a truly random way. For example, furnace failures are considered to be totally random. Their occurrences are equally likely at any logical time throughout the simulation.

An example of an event that is random within a known cycle time is removing cast bars from the casting equipment. Occurrence time for this task is assigned by a random distribution given a cycle time assigned by a known distribution.

A unit change on a casting unit is an example of a fixed time occurrence. The time for this task to occur was assigned before the simulation and its occurrence was forced to concur with this assignment.

It is worthy to note that the fixed occurrence tasks are in reality not fixed. They are referred to as fixed here, because they were dealt with in the actual simulation program as though they were fixed. These "fixed" occurrence times were preassigned by psuedo random generation. They were handled as "fixed" in the model to assure logical execution sequences.

As mentioned briefly before, each of the sixteen crane tasks were sub-divided by breakpoints into sub-tasks. This was done to facilitate the most realistic application of job practices.

Since many of the crane tasks are relatively long in duration and consist of several distinct steps, it was decided to allow the crane to be pre-empted at any breakpoint in a major task if a higher priority task was waiting. The pre-empted task or tasks would be completed as priorities and crane workload allowed. In this manner, the model operated the crane very much as it was physically operated. Queue statistics on all task and task breakpoints were tabulated for analysis.

The model was designed to simulate a month's activity. The smallest time unit used in the model was one minute. All time statistics were accumulative in terms of minutes.

An advantage capitalized on in this study was that initially, the simulation model simulated an existing set-up. Given the vast knowledge the manufacturing personnel had gained by experience, it was possible to calibrate or benchmark the accuracy of the model before adding the unknown element, the additional equipment. In this way, the manufacturing people gained additional confidence in the model, since they could see that it was simulating the existing operation with a high degree of accuracy.

BENCHMARKING THE MODEL

This phase proved to contribute more than any other to management's confidence in the simulation model and more importantly, the acceptance of its results.

Several simulation runs were made of the "before" situation. The results were tabulated for management review. Due to the structure of the model, it was possible to determine many significant operating parameters that could be measured against reality or known past performance. For example, it was possible to report production for the month, not only in total, but in some detail with respect to product mix. The fact that production generated by the model was totally believable, when compared to actual history, played a major role in convincing management that the model was valid. It was also possible to demonstrate that all prescribed tasks had been completed by the crane throughout the month. Again, task occurrences closely paralleled actual experience. (See APPENDIX D)

Given the confidence gained in the model by "benchmarking" it with reality, the additional equipment was added to the model for evaluation.

Needless to say, the model was designed so that the additional equipment could be easily incorporated. The author's intention and approach from the beginning was to model the present situation, gain the confidence of management by demonstrating the model's accuracy, and insert new equipment and measure its effect. This approach was taken to maximize the probability of acceptance of the studies results. The method proved to be a good approach.

ADDITION OF PROPOSED EQUIPMENT TO MODEL

The equipment that was projected to be added to the casting operation was incorporated into the existing and accepted model of the present casting operation.

Simulation runs were made with the revised model. By comparing the "before" and "after" simulation run results, it was possible to achieve the project objectives.

The confidence that had been gained in the model by this time really paid off throughout the remainder of the project.

The comparisons of before-and-after run results allowed predictions to be made on the critical characteristics of the crane operating system. Specifically, it enabled the estimation of increased casting delay due to crane task interference increases.

CONCLUSIONS AND RESULTS

Conclusions were drawn by comparing the simulation of the present casting operation, an accepted valid picture of reality, to the simulation results of the proposed casting operation with the additional equipment.

The major conclusion drawn showed that due to increased crane task interference, a 2.8% loss in capacity could be expected. This percentage was, in turn, converted into total pounds of production loss to be expected. This fact then became the major contribution of the simulation to the decision process. The magnitude of the problem had been quantified and it was now up to management to determine what action was required.

The important contribution this study made was that management was in a position to make a much more informed decision that it would have been without the simulation. Critical to this fact was that management accepted the results of the simulation as valid.

Due to the proprietary nature of many of the statistics involved, relative changes between present and proposed operations are shown; no absolute statistics are given.

Realizing the detailed structure of the model, it is possible to conclude that the difference measured between the two situations are attributable strictly to the increase in crane task interference. The following table shows some critical relative statistics between the two situations simulated.

COMPARATIVE STATISTICS BETWEEN PRESENT AND PROPOSED SITUATIONS SIMULATIONS

	PROPOSED SITUATION
Average Number of Casting Cycles Per Day Per Machine	2.8 % Decrease
Crane Workload	15 % Increase
Zero Wait Time Bar Removals	17 % Decrease
Average Wait Time For All Bar Removals	44 % Increase
Average Wait Time for Bar Removals That Had To Wait	No Change
Percent of Bar Removals When More Than One Production Unit Was Ready To Have Bars Pulled At The Same Time	7.9 % Increase
Percent of Occurrence of Bar Removals That Exceeded Allowable Delay	4 % Increase

SUMMARY

If this summary had to be limited to two words, those two words would be MANAGEMENT INVOLVEMENT.

Any competent O.R. oriented person can develop a simulation model of a situation. This fact alone, however, means very little. Unless the simulation gains management's confidence and acceptance, the study is worthless to the organization.

It is possible to greatly enhance the probability of management acceptance of a simulation if management involvement and participation in the simulation are forced to a maximum.

This involvement should spread over all levels of management involved and should be sustained throughout the study.

The final acceptance and utilization of a simulation study are less related to the technical excellence of the study than they are to the confidence it is possible to gain from management if the project is executed properly. This statement is not intended to say that technical excellence is not a requirement for ultimate success, but to stress that management participation cannot be overlooked or underestimated if simulation's full potential is ever to be realized.

APPENDIX A

DETAILED DESCRIPTIONS

TASK	DESCRIPTION	BREAKPOINT TIMES * (MINUTES)	PRIORITY
D. C. Casting Bar Removal	Cast bars must be removed from casting units and moves to other equipment.	11 (No Breakpoints)	10 (Top)
Coil Change	As a result of failure, heating coils must be replaced.	10-5-20-40-20-5-10	9
Replace Charge Weigh Line	During charge weigh line failure, crane must move materials to casting floor ordinarily done by charge weigh line.	6	8
Replace Gantry Crane	During gantry failure, crane must handle gantry crane's jobs on the casting floor.	6	8
D.C. Mold Carriage Change	Replacing the mold carriage on D.C. Casting units.	6-6-6-6	7
D.C. Mold Liner Change	Replacing the mold liners on D.C. Casting units.	5-5	7
D.C. Unit Change	Completely changing 5 melt furnaces and holding furnace associated with D.C. Casting	145-10-10-10-10-10-10-10-10-60-10-10-10-10-120-6-6-6-6-10-10-10-10-120-20-15-30	6
Unplanned Melt Changes	Changing only 1 of the 5 melt furnaces on a casting unit as the result of some failure.	10-5-10-40-10-15-20	6
5-Melt Change	Replacing the 5 melt furnaces assocated with	10-5-10-40-10 Then 13-10 5 Times 5 Times	6

DETAILED DESCRIPTIONS

TASK	DESCRIPTION	BREAKPOINT TIMES * (MINUTES)	PRIORITY
Random Tasks	Small jobs such as moving materials to casting floor.	5	5
Ascast Bar Removal	Removing bars from another type of casting unit.	5	5
Booked Mold Casting Mold Changes	Replacement of molds on "book mold" casting unit.	20-25	4
Ascast Mold Change	Replacement of molds on "ascast" casting unit.	6-6	4
Ascast Furnace Change	Replacement of melting furnace on ascast unit.	15-15-10-10-10	4
Changing Wertli	Molten metal must be carried to this production unit by the crane.	10	4
Wertli Failures	Given certain types of failures on this production unit, the overhead crane is required to help recover the unit.	10-20-10	4

* Underlined times are times crane is required. Times not underlined must elapse before the next step is done, however, the crane is free to do another job during the non-underlined times.

APPENDIX B

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1. D.C. Bar Removal	X	X			X	X	X		X							
2. Coil Change	X	X					X	X	X							
3. Charge/Weigh Failure			X													
4. Gantry Failure				X												
5. D.C. Mold Carriage	X				X		X									
6. D.C. Mold Liner	X					X	X	X								

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
7. D.C. Unit Change	X	X			X	X	X	X								
8. Unplanned Melt Chg.		X				X	X	X	X							
9. 5-Melt Change	X	X						X	X							
10. Random Task										X						
11. Ascast Bar Removal											X		X	X		
12. Book Mold Mold Chg.												X				
13. Ascast Mold Chg.											X		X			
14. Ascast Fce. Chg.											X			X		
15. Charge Wertli															X	X
16. Wertli Failure															X	X

Note: X Indicates Row Event and Column Event May Not Occur Simultaneously.

APPENDIX C

DESCRIPTION OF MANUFACTURING OPERATION

This is a large manufacturing business producing copper and copper-base alloy strip. Specifically of interest here is the casting operation.

The casting operation employs a semi-continuous direct-chill casting process. The function of the casting operation is to provide large cast bars of non-ferrous alloys for further processing by rolling mills and fabricators. The bars are cast from various combinations of virgin and scrap raw materials.

The casting operation consists of a number of sub-systems which include melting furnaces, holding furnaces, casting pits, and various supporting material handling equipment.

Perhaps the most critical piece of material handling equipment is the overhead cranes which has the ability to move the entire length of the casting plant. The crane tasks are described in detail in APPENDIX A.

The crane has serviced the operation in a totally adequate manner to-date. Now, with plans to add casting equipment, the question arises as to whether the crane will be able to handle the subsequent increase in crane tasks.

APPENDIX D

COMPARISON OF HISTORICAL AND SIMULATED TASK FREQUENCIES

	% ERROR IN SIMULATED FREQUENCY
1. D.C. Bar Removal	Not Applicable
2. Coil Changes	0%
3. Charge/Weigh Failures	30%
4. Gantry Failures	30%

COMPARISON OF HISTORICAL AND SIMULATED TASK FREQUENCIES

	% ERROR IN SIMULATED FREQUENCY
5. D.C. Mold Carriage Change	0%
6. D.C. Mold Liner Change	0%
7. D.C. Unit Change	0%
8. Unplanned Furnance Changes	40%
9. 5-Melt Changes	0%
10. Random Tasks	4%
11. Ascast Bar Removal	20%
12. Book Mold Mold Changes	0%
13. Ascast Mold Change	0%
14. Ascast Furnace Changes	0%
15. Charging Wertli	20%
16. Wertli Failures	10%

APPENDIX E

PROGRAMMING CONSIDERATIONS

The simulation model was written using RCA's Flow Simulator language. Flow Simulator is a language very similar to G.P.S.S. and was selected basically because the computer available was a RCA Spectre 70 on which the simulation language available was Flow Simulator.

The programming problem most basic to the success of the project was, of course, representing the crane operating system in the most realistic manner possible. To achieve this goal, certain types of activities were fixed in time prior to the simulation run. This pre-assignment was done using a combination of randomness and logical pattern knowledge for events occurrences.

In the simulation program, these pre-assigned tasks were handled as follows: A function was set for each day to be simulated which contained code for the proper pre-assigned tasks to take place on that day. These functions, in conjunction with the various generate statements, kept transactions coming at the crane in a logical pattern throughout the simulation.

Both tasks associated with pre-assigned events and random events contained breakpoints as explained in the body of this report. These breakpoints allowed the crane to be pre-empted within a job, if a higher priority task came due. The task time distribution associated with the various tasks were handled by creating a function for each task which included advance times for service and free time for the crane within a job execution.

For example, a job could consist of the following -

5 Minutes	Crane Time
6 Minutes	Crane Time
10 Minutes	Non-Crane Time
7 Minutes	Crane Time

During the 10-minute non-crane time, the crane was considered to be free for executing other tasks.

Various tables were set up to collect data in addition to the standard data output of Flow Simulator. For example, a table was generated for assigned cycle times and for realized cycle times. Comparing those two distributions was helpful in demonstrating the effect of workload on casting cycle time realization.

Tables were also kept showing interference patterns. These tables indicated what tasks were interferring with what other tasks. This information was collected to help in designing new procedures for the actual crane operating system to minimize interference in the future design.

The smallest unit of time considered in the simulation was one minute; the total simulated time period was one month. The simulation took approximately 25 minutes to simulate one month and required 220K of core.

CYCLE-TIME SIMULATION FOR A MULTIPRODUCT MANUFACTURING FACILITY

M. M. Patel, J. M. Panchal, and M. T. Coughlin
IBM System Products Division, East Fishkill
Hopewell Junction, N. Y. 12533

Abstract

This is a generalized simulation model for a multiproduct manufacturing line with interdependent production equipment. Based on various product demands, it simulates resources such as manpower and equipment and generates product cycle time. It is a deterministic model. It takes into consideration equipment reliability, man-machine interactions, yields, rework, and process-related constraints. The model could also be used to plan resource requirements to fulfill required product cycle times. The model is written in GPSS language with PL/1 subroutines. The temptation to include relatively less pertinent factors is resisted in order to keep the model economical.

INTRODUCTION

The successful commitment of resources is one of the most crucial responsibilities plant management faces. This is particularly true in the semiconductor business, where new products are introduced at an ever-increasing pace from laboratories, and new product applications create modifications of current products. These factors -- and the spur of competition -- make product manufacturing cycle time more important than ever.

The cycle time to manufacture a product is affected by three major factors: equipment, human resources, and buffer or work in process. The dynamic nature of business is very complex, however, due to many variables: demand fluctuations, change in product mix, addition of new products, variation in number and types of manufacturing operations for different products, variation of process times for operations from several minutes to several hours, batch type and individual unit operations, and variation of batch size by operation.

To gain insight into such complexities and an understanding of interrelationships is beyond the capability of one person without the aid of some meaningful tool. Simulation is one of the most exciting techniques employed. When enhanced by high-speed computers, simulation makes it possible to tackle complex problems in very short times. It also deals with dynamics. The real-world complexities can be closely represented in a physical model on paper. Manipulation of the model for different strategies enables management to find out what the probable results would be, and thus leads to making sound business decisions before actually committing the resources.

This paper describes a fairly complex, deterministic simulation model of a multiproduct manufacturing facility. The model was developed primarily to better understand the interrelationships of the complexities of the manufacturing floor and to reduce the product manufacturing cycle time. Such insight permits faster manufacturing response and a reduction in time and cost when introducing product changes. An independent, controlled experiment carried out on the manufacturing floor helped to validate the model by achieving results on the floor. The combined efforts of model and experiment led to a reduction in product queueing times by 66 percent.

Depending on the basic information and the end result sought after, the model can give product cycle time based on the available resources, or the resources required to meet the planned cycle times. The impact of a proposed engineering change or group of changes can be meaningfully analyzed. The model is not designed to give the global optimum solution in one exercise, but can lead to a nearly optimal solution by iteration. The economics in terms of dollars and cents has to be evaluated externally for the different iterations.

The model enables management to analyze the impact of the following manufacturing parameters: 1) daily schedule start by product, 2) product route and modifications, 3) rework loop or loops, if any, and different rework percentages of virgin products, 4) equipment plans, 5) manpower and productivity, 6) job enlargement, 7) production losses or yields by operation or group of operations, 8) operation-level time parameters, i.e., unit or batch process time, 9) reliability and maintainability of equipment, and 10) plant operation policy for shifts, working days per week, etc.

The model is a generalized one to simulate most discrete manufacturing facilities. It is designed to provide a general framework for simulating the flow of jobs through a manufacturing facility using specified equipment, manpower, and buffer (WIP) resources. The model is written in GPSS-V language with PL/1 subroutines. The PL/1 subroutines are used for two purposes: for ease of input data manipulation, and for GPSS output report summarization and interpretation. A typical system requirement for ten products having a routing of 90 to 300 operations and about 350 pieces of equipment can be about 400K bytes of core space. To simulate two months of manufacturing activity for the situation just described can take about 25 min of CPU time on a System/360 model 85 computer. It is essential that the simulation time be long enough to reach steady-state conditions.

The outline of the model structure and its salient features are presented here.

MANUFACTURING CYCLE TIME

The cycle time of a product consists of the actual time to process for each operation, the waiting time before each operation, and the transfer time between operations.

Reliability of the equipment also has an impact on the cycle time and buffer requirements. To maximize utilization of equipment and people usually requires large buffer volumes, which leads to longer cycle time. Immediate implementation of a mandatory product change becomes very costly because of the scrapping of large buffer volumes.

Figure 1 presents a sketch of a unique manufacturing plant segment. Say there are 5 products, 8 operations, and 32 pieces of equipment, within 12 equipment groups. Let A-E represent the products and 1-8 represent the operations. The schematic shows the product dedication by equipment group for each operation, as well as the rework path. Any discrete manufacturing facility could be represented similarly by a schematic drawing for simulation studies.

MANUFACTURING LINE REPRESENTATION

Figure 2 represents the system and logic modules, which are described below.

Facility Representation

Physical realities of a manufacturing plant are closely represented in the model by equipment groups by departments or segments of a department.

Equipment Parameters

Each equipment group can be given two parameters, for planned and unplanned downtime or maintenance. The planned downtime can be specified as a fixed-time activity for one of the periods, such as each shift, each day, or each week, etc. By scheduling downtime, it is possible to reflect start-up activity, periodic instrument calibration, etc. Unplanned maintenance can be specified by two factors: mean time between failure (MTBF) and mean time to repair (MTR). Each factor can be either of fixed time value or a mean value of a statistical distribution with its variance.

Manpower

Manpower is assigned by a manpool, which provides service to given equipment groups based on skill and training. The man-machine relationships, such as one man per machine, one man for more than one machine, and more than one-man crew per machine, can be specified. Here, job enlargement policies to eliminate monotonous activities can be studied.

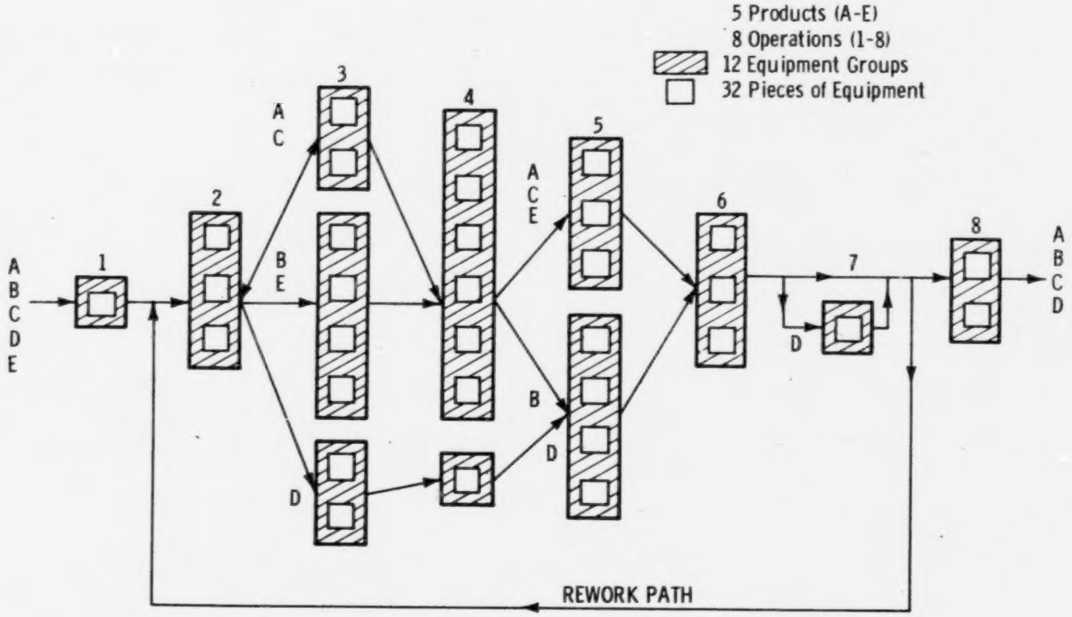

5 Products (A-E)
8 Operations (1-8)
12 Equipment Groups
32 Pieces of Equipment
A
B
C
D
E
1
2
3
A
C
B
E
D
4
5
A
C
E
B
D
6
7
D
8
A
B
C
D
REWORK PATH

Fig. 1. Schematic of a unique factory segment.

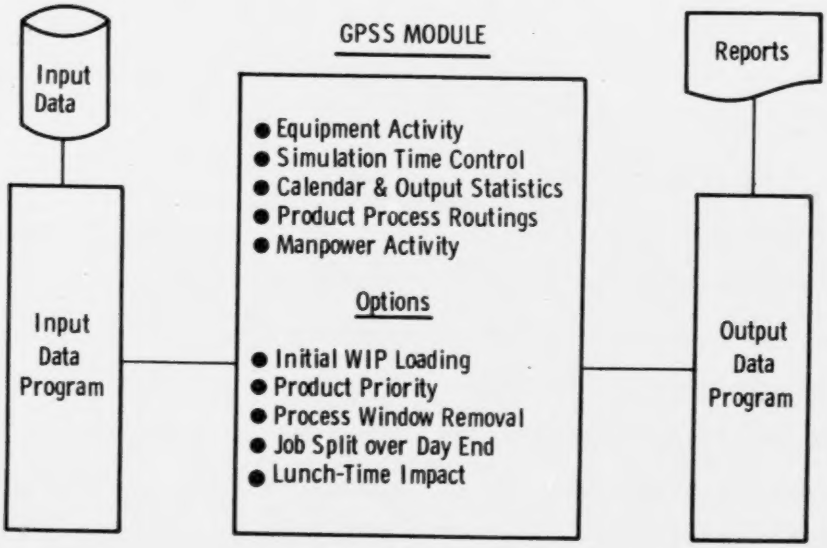

Input Data
Input Data Program
GPSS MODULE
● Equipment Activity
● Simulation Time Control
● Calendar & Output Statistics
● Product Process Routings
● Manpower Activity
Options
● Initial WIP Loading
● Product Priority
● Process Window Removal
● Job Split over Day End
● Lunch-Time Impact
Reports
Output Data Program

Fig. 2. System and logic modules.

Staggered lunch periods for certain operation coverage can be realized by using a cafeteria log algorithm. For example, where cafeteria service is available between 11 a.m. and 1 p.m. for the morning shift and the lunch period is for one-half hour, an operator would go to lunch within the cafeteria service time depending on the product flow and the process time for a given operation.

Process Routings

Each product has detailed process routings for sequential virgin operations and rework paths, as required. Two types of rework paths can be specified, as shown in Fig. 3. The rework is given as a percentage of the jobs for each path. Also, the maximum number of times a job can be reworked at an operation can be specified.

Operation Parameters

Each operation can be given three time parameters, two for process time (A and B) and the third for manpower (C). The use of two process parameters permits a single-step or multiple-step process to be represented. A single-step process, such as inspection, testing, baking, milling, etc., is shown in Fig. 4. Multiple-step processes such as chemical clean-rinse-dry, progressive drawing, bearing cage stamping, multiple-pass grinding, etc., can be represented as shown in Fig. 5. In such cases, parameter A is the time interval for a unit or batch to enter the system, and parameter B is the remaining time a unit or batch has to spend in the system.

An outline of the basic job flow is shown in Fig. 6.

Work-Time Policy

The work time policy can be specified for the following:

- Working hours per shift.
- Number of shifts per day.
- Number of working days per week.
- Number of weeks in a period.
- Number of periods to be simulated.

Process Window

The program takes into consideration whether a job can be finished before the end of the working day. If the available time is less than the process time, then the job waits unitl the next day or is processed on overtime.

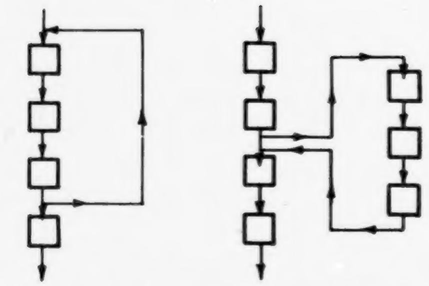
MODEL OPTIONS

Buffer/WIP

Initial buffer distribution by process step by product can be assigned as an input or the model can be run without it.

Product Priority

If desired, priority can be assigned to a product or group of products. Otherwise, the model will handle buffer and the released products on a FIFO (first-in, first-out) basis.



(a) Internal Path (b) External Path

Fig. 3. Rework paths.

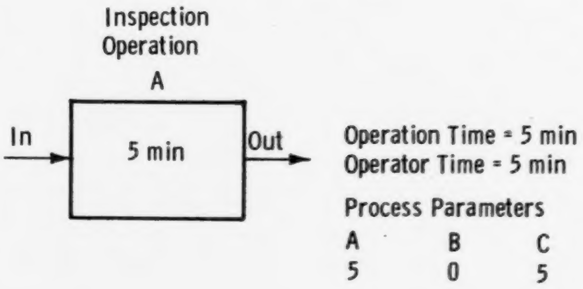

Inspection
Operation
A
In
5 min
Out
Operation Time = 5 min
Operator Time = 5 min
Process Parameters
A B C
5 0 5

Fig. 4. Single-step operation.

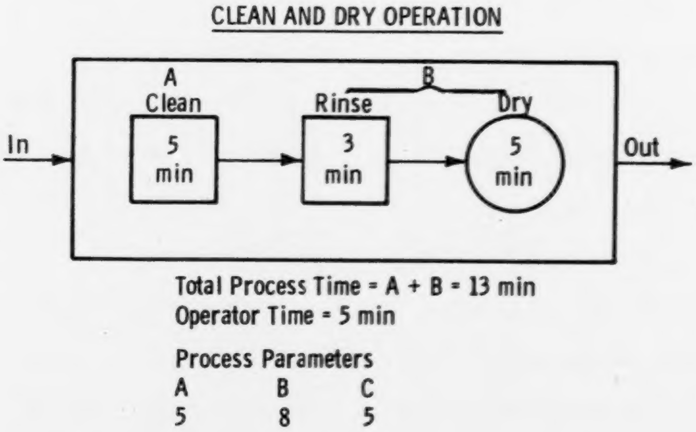

CLEAN AND DRY OPERATION
A
Clean
B
Rinse
Dry
In
5
min
3
min
5
min
Out
Total Process Time = A + B = 13 min
Operator Time = 5 min
Process Parameters
A B C
5 8 5

Fig. 5. Multiple-step operation.

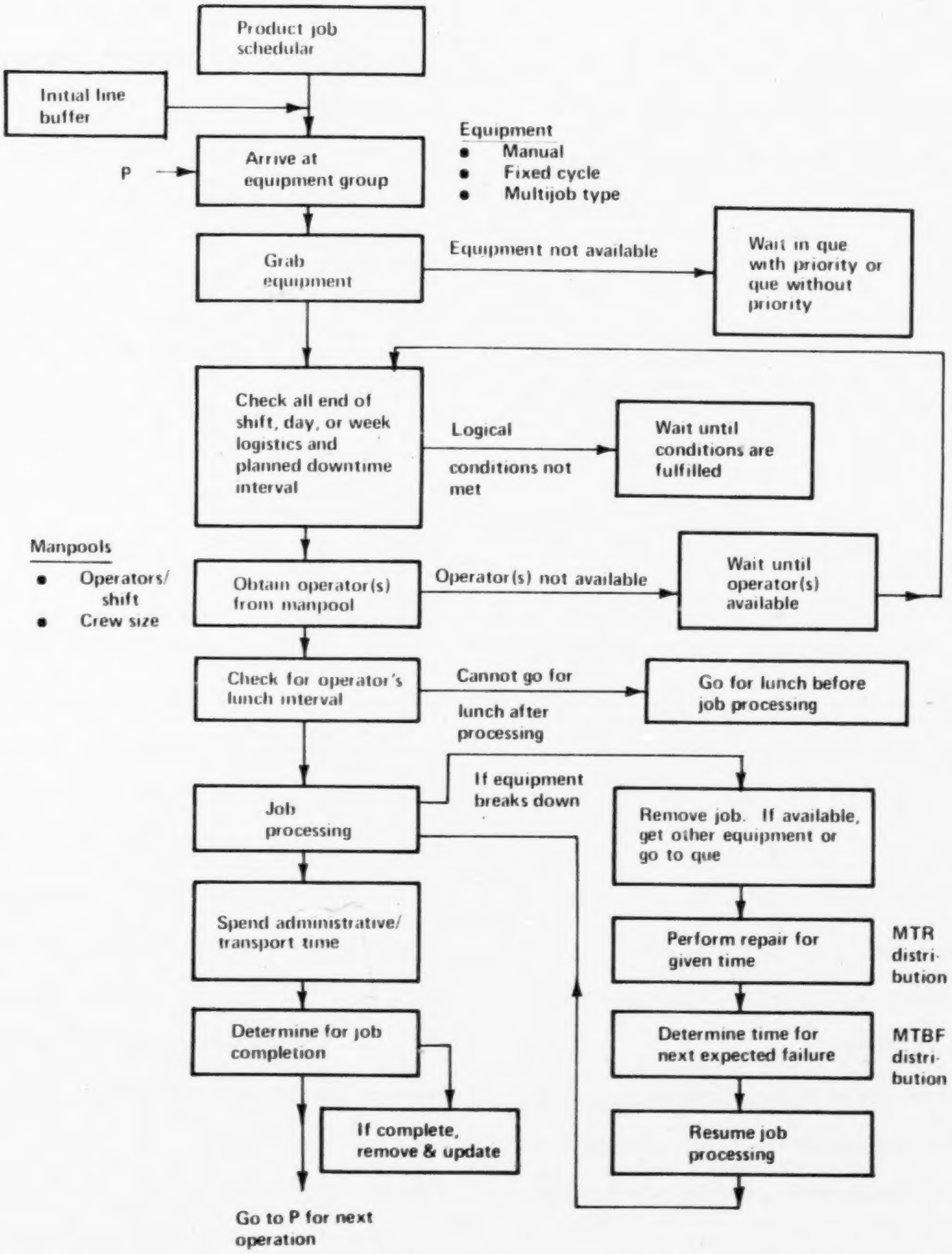

Product job schedular
Initial line buffer
P
Arrive at equipment group
Equipment
Manual
Fixed cycle
Multijob type
Grab equipment
Equipment not available
Wait in que with priority or que without priority
Check all end of shift, day, or week logistics and planned downtime interval
Logical conditions not met
Wait until conditions are fulfilled
Manpools
Operators/shift
Crew size
Obtain operator(s) from manpool
Operator(s) not available
Wait until operator(s) available
Check for operator's lunch interval
Cannot go for lunch after processing
Go for lunch before job processing
Job processing
If equipment breaks down
Remove job. If available, get other equipment or go to que
Spend administrative/transport time
Perform repair for given time
MTR distribution
Determine for job completion
Determine time for next expected failure
MTBF distribution
If complete, remove & update
Resume job processing
Go to P for next operation

Fig. 6. Basic job flow.

Process Window Removal

On equipment with uninterruptable process time, this option enables a plant to get one extra batch a day per piece of equipment specified. Maximum overtime in such cases would be equal to the 'process time' by processes by pieces of equipment.

Lunch Time

Depending on the manufacturing environment, this option can be exercised or not in the simulation model.

Job Split

This is useful when a product is released in a lot size of 50, 100, or any number of units, and in some operations performed on a unit basis. Use of this option permits partial processing of a lot at the end of a working day and the remainder on the following working day.

Manpower

The input subroutine program can compute manpower by pool by shift based on the release schedule, or it can be preassigned in the input data.

OUTPUT REPORTS

The contents of the output reports are designed to give only pertinent data to facilitate decisions for all levels of management. They are enumerated as follows:

1. Total jobs and quantities released at the first operation, and jobs and quantities placed in stock by period by product.
2. Sector or segment and over-all cycle-time mean with standard deviation, and sector raw process time (i.e., sum of each operation's process time in the sector) by product.
3. Equipment group utilization and maximum queue buildup.
4. Manpool and over-all manpower utilization.
5. Buffer/WIP distribution in line by operation by product.

CONCLUSIONS

The simulation system described in this paper can be used as a tool to aid plant management in evaluating alternative strategies, where selection of equipment, manufacturing processes, new facility design, or modification of existing facility is involved and short cycle times are required. The immediate management responsible for production can visualize the impact of job enlargement, process step modifications and/or eliminations, different levels of manpower and equipment utilizations, and product volumes, on cycle time.

ACKNOWLEDGMENTS

The authors thank the engineering support group for contributing actual data, and express their gratitude to Mr. J. D. Peruffo and Mr. S. A. Rolleri for their support and encouragement.

Session 7: Urban Problems
Chairman: Gary Brewer, The RAND Corporation

The panel examines several operational uses of simulation in the urban problem solving environment. Papers range from an attempt to develop a generalized urban planning model capable of managing substantive areas as varied as health and education, all the way through to a very specific example of a police patroling dispatching simulation. Other examples include an effort to understand, model and improve a portion of New York's troubled judicial system and a deployment model used by the New York Fire Department. The emphases throughout are serious and operational. Discussants have been chosen not only because of their technical qualifications to comment on model specification, construction and operation but also because they each have had considerable experience dealing with the specific substantive issues treated in the discussed simulations.

Papers

"A Demographic Simulation Model for Health Care,
Education, and Urban Systems Planning"
Philip F. Schweizer, Westinghouse Electric Corporation

"Simulation Model of New York City's Felony Adjudicatory System"
Lucius J. Riccio, Lehigh University

"A Simulation Model of the New York City Fire Department:
Its Use as a Deployment Tool"
Grace Carter, Edward Ignall and Warren Walker,
the New York City RAND Institute

"On-line Simulation of Urban Police Patrol and Dispatching"
Richard C. Larson, Massachusetts Institute of Technology

Discussants

Daniel Alesch, The RAND Corporation
John Jennings, The New York City RAND Institute
Gary Brewer, The RAND Corporation
James Kakalik, The RAND Corporation

A DEMOGRAPHIC SIMULATION MODEL FOR HEALTH CARE, EDUCATIONAL, AND URBAN SYSTEMS PLANNING

Philipp F. Schweizer
Westinghouse Research Laboratories
Pittsburgh, Pennsylvania 15235

Abstract

This paper describes the development, application, and digital computer simulation of a demographic model suitable for long term planning. The simulation model is based on the "cohort survival methodology" and projects population characteristics (population numbers for each region, age group, sex, year and racial or income group) for a planning period less than or equal to twenty years.

Demonstration of how the simulation model is applied to problems in Health Care, Educational and Urban Systems Planning are presented.

1. Introduction

This paper discusses the development and application of a demographic model for use in long term planning.

The need for demographic information became apparent from previous work concerned with the development of planning tools for the Westinghouse Health Systems Department and the Westinghouse Learning Corporation. Population forecasts were necessary to determine the future demand on health care facilities and to predict enrollment in planning educational facilities.

An initial investigation was conducted to determine the demographic information available from local planning groups or the U.S. Census Bureau and whether or not this information was sufficient for the intended planning purposes.

An examination of the information available from the U.S. Census Bureau uncovered the following difficulties: 1) Population forecasts for the U.S. and most states were available, however, forecasts for counties and local areas were rare. When and where these forecasts for local area existed, quite often, they were merely ratios of aggregate projections for the state or county. The forecasts did not account for local influences on the population. 2) The population projections are commonly given for 10 or 15 year periods. However, many planners must make decisions on a yearly or even monthly basis. In an area where rapid change prevails interpolation of the projections may be difficult. 3) In most cases a single population forecast was given which did not yield a sensitivity of the population to various factors such as fertility rates, mortality rates, migration, employment, housing development, etc. This sensitivity information is a valuable asset to the planner and could be provided with a family of forecasts, however, these again when provided were usually with respect to a single factor. Recognizing that the factors of interest to various planners would be different, a scheme was needed for producing forecasts which were a function of local influencing factors.

In studying information from some local planning groups, it was apparent that the quantity and quality of available information was highly variable and depended on the size and sophistication of the specific planning group. Since the program goal was to provide a planning tool that would be applicable to any local area or state it was assumed to be too risky to rely on a local planning group for sufficient information.

The conclusion drawn from the above was that although the Census Bureau could provide regional population forecasts and local planning groups could provide many of the "necessary bits and pieces" of the demographic picture, this information alone would not be sufficient for local and even state planning needs. A demographic model would be necessary to manipulate this data into a more usable form.

The next stage of the investigation involved the resolution of whether a demographic model should be developed or whether an existing model could be used.

Models operating on past history were available (Ref.5). Significant work in the area of statistics has allowed planners using regression techniques to build models based on past history alone. The projections from this type of model have provided valuable information for the short term. Caution certainly must be exercised in applying these models for long term planning. In many cases even though the short term projections are accurate, the long term projections are misleading. A better approach appeared to be a technique which made use of past history but also weighed the planner's subjective judgments about the future (i.e., housing development, land use, future employment).

Investigating various available models (Refs. 4,5,6) showed that some were aimed at very general studies (i.e., population forecasts for U.S., the world, India, China, etc.) while others at very specific applications. None of those uncovered seemed directly appropriate to the problem of forecasting population for a rapidly developing suburban area or new town which was one of the primary intentions of this work. It was concluded that a model should be developed since modification of existing models would require as much or more effort.

The approach finally taken was one based on the "cohort survival method" which has been adopted for use by the U.S. Census Bureau. In general, the "cohort survival method" begins with the detailed distribution of a population obtained in a base year (most likely a census year), and moves that population through time applying to it various population changing factors, according to a set of assumptions about those factors. A model was constructed using this methodology but with modifications for including local population influencing factors.

Refs. 1 through 4 provide background information for the decisions made and the models developed in this study. Although none of the information from these references were explicitly used, considerable insight for the modeling problem was obtained from them.

The remaining sections of this paper present the mathematical model (Mathematical Model Description), discuss the computer code used in implementing the model (Computer Code), and demonstrate applications (Applications) of the simulation model.

2. Mathematical Model Description

The model classifies the total population according to five factors: 1) geographical location (region), 2) age groups, 3) year of existence, 4) race or income group, and 5) sex. A variable (or state) is assigned to represent the number of people possessing any possible combination of the above five factors.

The total population is divided into age groups or cohorts. Diagrams depicting the four basic considerations, aging, mortality, fertility and migration, that are modeled for each cohort are shown in Figures 1 and 2.

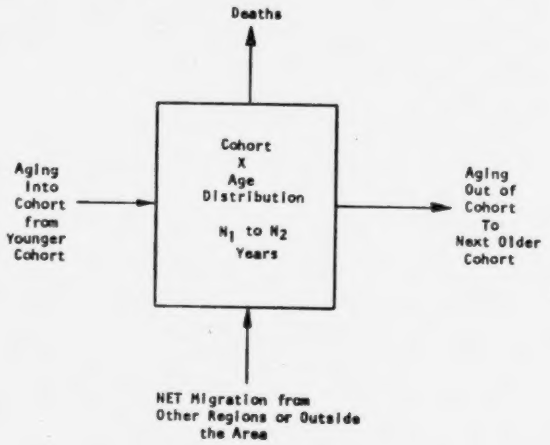

Deaths
Aging Into Cohort from Younger Cohort
Cohort X Age Distribution
N_1 to N_2 Years
Aging Out of Cohort To Next Older Cohort
NET Migration from Other Regions or Outside the Area

Fig. 1—Single Cohort (age group) model

The dynamics are incorporated in the model by changing the number of people that belong to the cohort each planning period (usually each year). A certain number of people are removed from the cohort to represent those that have aged to the next older cohort, those that have died during the planning period and those that have migrated from the area. Numbers of people are added to the cohort to represent those that are aging from a younger cohort and those that are migrating into the area.

The above modeling procedure may be mathematically expressed for each cohort by the following expression.

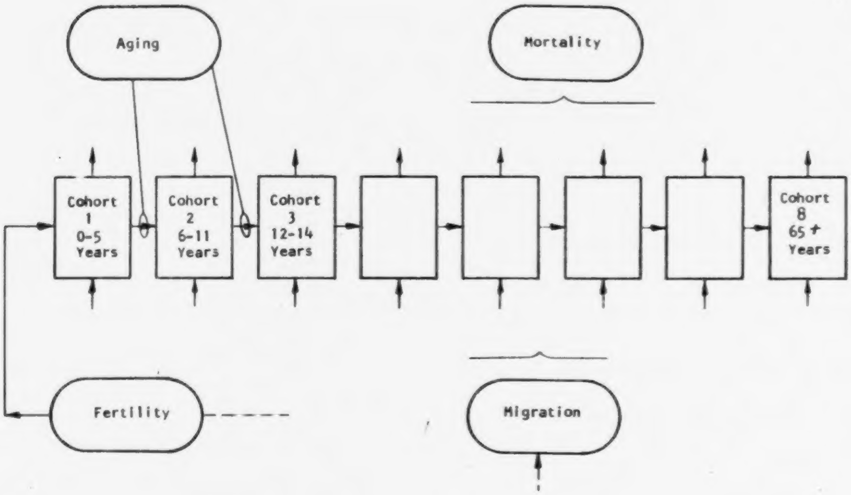

Aging
Mortality
Cohort 1 0-5 Years
Cohort 2 6-11 Years
Cohort 3 12-14 Years
Cohort 8 65 + Years
Fertility
Migration

Fig. 2—Functional diagram of Demographic Model depicting the four primary considerations

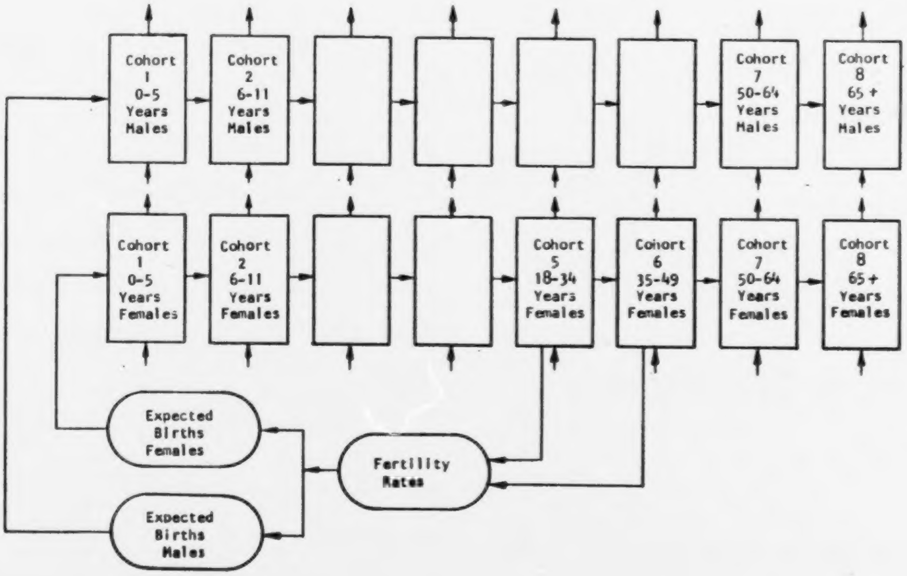

Cohort 1 0-5 Years Males
Cohort 2 6-11 Years Males
Cohort 7 50-64 Years Males
Cohort 8 65 + Years Males
Cohort 1 0-5 Years Females
Cohort 2 6-11 Years Females
Cohort 5 18-34 Years Females
Cohort 6 35-49 Years Females
Cohort 7 50-64 Years Females
Cohort 8 65 + Years Females
Expected Births Females
Fertility Rates
Expected Births Males

Fig. 3—Functional diagram depicting the feedback effect of female population through expected births

$$x^{m,n}_{i,j,k+1} = (1-A^{1}_{j,k})\, x^{m,n}_{i,j,k} + A^{2}_{j,k}\, x_{i,j-1,k} + y^{m,n}_{i,j,k} + F^{m} \sum_{j=JCB1}^{JCB2} \alpha_j\, x^{m,1}_{i,j,k} + M^{m,n} D_j^{\,n} \sum_{\substack{i,j,m\\n}} x^{m,n}_{i,j,k} + H_{j,p}\, v_{i,p,k} \qquad (1)$$

where

$x^{m,n}_{i,j,k}$ represents the number of people in a particular population cohort

with subscripts

- i designating the demographic region
- j designating the age grouping of the cohort
- k designating the year.

and with superscripts

- m designating the race or income group
- n designating the sex.

Each term in Equation (1) will subsequently be discussed with regard to its contribution to the total expression.

The first term, $(1-A^{1}_{j,k})x^{m,n}_{i,j,k}$ represents the difference between the cohort population in the k^th year and the number of people who will leave this cohort during the k^th year because of aging.

The aging parameter $A^{1}_{j,k}$ is computed from

$$A^{1}_{j,k} = CD_{j,k}/CS_{j,k} \qquad (2)$$

where

$CS_{j,k}$ is the span of cohort j in the k^th year

$CD_{j,k}$ is determined from the age distribution of cohort j in the k^th year.

The second term, $A^{2}_{j,k}x^{m,n}_{i,j-1,k}$ represents the number of people who will enter the cohort during the k^th year because of aging. The aging parameter $A^{2}_{j,k}$ is computed from

$$A^{2}_{j,k} = CD_{j-1,k}/CS_{j-1,k} \qquad (3)$$

where terms in Equation (3) are as defined in Equation (2).

The third term $y^{m,n}_{i,j,k}$ represents the migration into or out of the i^th region for the j^th cohort and k^th year.

The fourth term, $F^{m} \sum_{j=JCB1}^{JCB2} \alpha_j\, x^{m,1}_{i,j,k}$ models the expected births during the k^th year. The parameters used in this term are defined by

F^{m}	the fertility rates for each race, m
JCB1	the first cohort with women of child-bearing age (15-44 years)
JCB2	the last cohort with women of child-bearing age
α_j	fraction of the cohort of child-bearing age.
$x^{m,1}_{i,j,k}$	female population in cohort j, region i, year k, and race m.

Figure 3 depicts the effect of this fertility term on the total population. Females in the child-bearing cohorts are multiplied by fertility rates to determine expected births for males and females.

The fifth term, $M^{m,n}D^{n}_{j} \sum_{\substack{i,j\\m,n}} x^{m,n}_{i,j,k}$ models the expected deaths during the k^th year (Fig. 4). The parameters of this term are defined by

$M^{m,n}$	the mortality rates for each race and sex
D^{n}_{j}	the mortality distribution (the portion of total deaths that are in the j^th cohort with sex n)

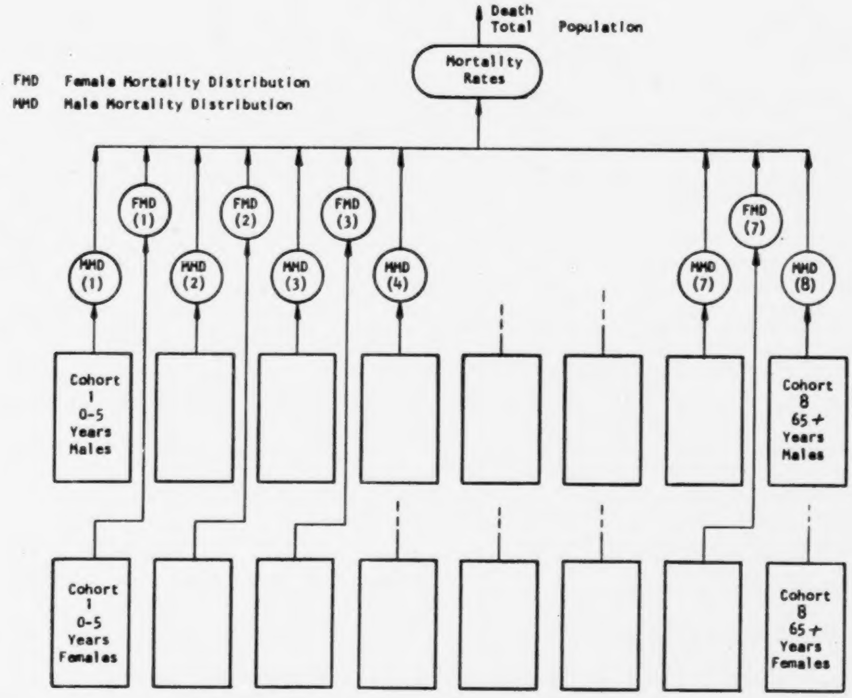

Death
Total Population
Mortality Rates
FMD Female Mortality Distribution
MMD Male Mortality Distribution
FMD (1)
FMD (2)
FMD (3)
FMD (7)
MMD (1)
MMD (2)
MMD (3)
MMD (4)
MMD (7)
MMD (8)
Cohort 1 0-5 Years Males
Cohort 8 65+ Years Males
Cohort 1 0-5 Years Females
Cohort 8 65+ Years Females

Fig. 4—Functional diagram showing the interaction of mortality rates and distribution with each cohort

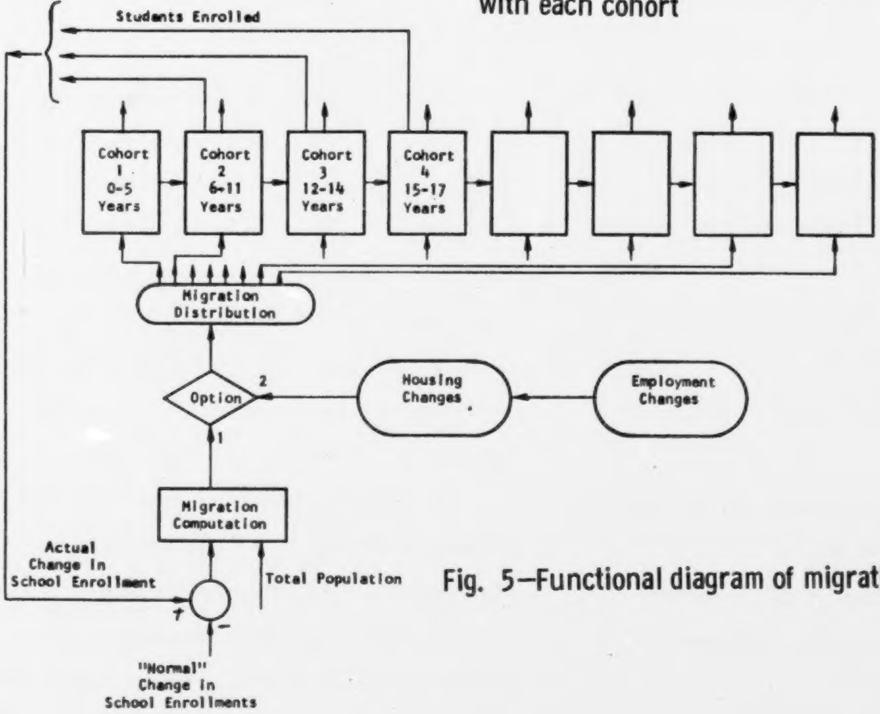

Students Enrolled
Cohort 1 0-5 Years
Cohort 2 6-11 Years
Cohort 3 12-14 Years
Cohort 4 15-17 Years
Migration Distribution
Option
2
1
Housing Changes
Employment Changes
Migration Computation
Actual Change in School Enrollment
Total Population
+
−
"Normal" Change in School Enrollments

Fig. 5—Functional diagram of migration model

$\sum_{\substack{i,j \\ m,n}} x^{m,n}_{i,j,k}$ represents the total population (the summation over all regions, cohorts, races, and sex).

The last term in Equation (1) models the effect of the population because of changes in types and numbers of dwelling units. The coefficients $H_{j,p}$ represent the average number of people belonging to cohort j and living in a dwelling unit of type p. The variable $v_{i,p,k}$ represents the number of p type dwelling units in region i in year k. These dwelling units could be considered as premium, choice, and economy houses; luxury, convenience and low income apartments; townhouses and mobile homes. The distinction between premium, choice and economy housing is made by considering the lot size and living space area. The distinction between luxury, convenience, and low income apartments is based on living space area. These classifications of dwelling units are actually arbitrary and may be changed by the planner if desired.

The scholastics defined as those students enrolled in kindergarten through grade twelve, are determined from the general population by the following expression

$$S_{i,r,k} = ED_{i,r,k}\, x_{i,j,k} \tag{4}$$

where

$S_{i,r,k}$ represents the scholastics in region i, grade r, and year k.

$ED_{i,r,k}$ is the fraction of the j^{th} cohort in grade r for region i and year k.

The migration term in Equation (1) may be known from other considerations or may be computed from the following expression

$$y^{m,n}_{i,j,k} = SECA^{m}_{i}\ (AFS/ANSAC)(x^{m,n}_{i,j,k} / \sum_{j=1}^{NC} x^{m,n}_{i,j,k} \tag{5}$$

with

$$SECA^{m}_{i} = (\sum_{k=1}^{NYEH} SEC^{m}_{i,k})/NYEH$$

where

$SECA^{m}_{i}$ = average school enrollment change over the past for region i, and race m

$SEC^{m}_{i,k}$ = actual school enrollment change for region i, year k, and race m

NYEH = number of years of school enrollment history

AFS = average family size

ANSAC = average number of school age children per family

NC = number of cohorts

The reasoning behind Equation (5) is as follows. First, the number of immigrating new school enrollments is estimated based on past history, SECA. This number is then divided by ANSAC to determine immigrating families and subsequently multiplied by AFS to determine the immigrating population. To allocate this population to appropriate cohorts, the ratio in Equation (5) is used.

The model for migration (Fig. 5) allows the planner to use one of two options (term 3 or 6 in Equation (1)). Option 1 uses Equation (5) and computes migration from a knowledge of past and present school enrollment changes by race, the average family size and average number of school age children. Option 2 uses the sixth term in Equation (1) to compute migration based on the type and number of dwelling units being constructed or removed from the area.

In cases where an exact number of future dwelling units are unknown the units for the planning period may be dynamically represented by

$$v_{i,j,k+1} = v_{i,j,k} \, C_i \qquad (6)$$

where

C_i is an estimated annual rate of change of dwelling units for region i.

When estimates of future employment are available, the rate of change, C_i is determined from

$$C_i = KE_i \qquad (7)$$

where

K is a proportionality constant

E_i represents the annual rate of change in basic employment in region i.

In making the population projections, the maximum population or saturation condition for each region must constrain the population numbers for each region. This constraints is enforced by the following expression.

$$0 \leq x^{m,n}_{i,j,k} \leq x^{m,n}_{SATi,j} \qquad (8)$$

where

$x^{m,n}_{SATi,j}$ is the maximum expected popuatlion in the j^{th} cohort region i for race m and sex n.

This saturation population for each region is computed from an assumed set of characteristics for the neighborhood. These characteristics include type and number of dwelling units, and the average number of people per type dwelling unit. The expression used in the computation is given by

$$x^{m,n}_{SATi,j} = (ANP_p)\ (PR_i)\ (TA_i)/ALS_p \qquad (9)$$

where

ANP_p represents the average number of people living in a p type dwelling unit for cohort j

PR_i represents the percentage of region i that will be devoted to residential development

TA_i is the total area in region i

ALS_p is the average area occupied by a p type dwelling unit.

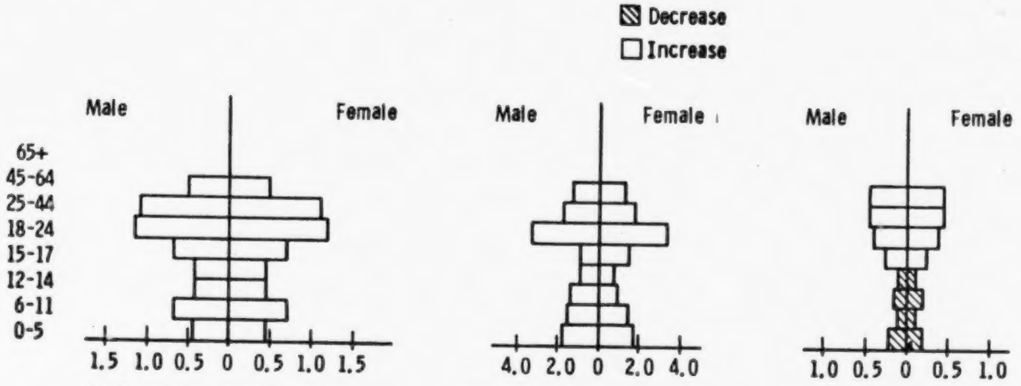
Equations (1) through (9) are used to make the population charactieristics projections for the planning period.

3. Computer Code

The model equations presented in the preceding section have been implemented in a computer code which is written in Fortran V and is operational on the Univac 1106 computer at the Westinghouse Research Laboratories. A version of the code in Fortran IV is also available at other Westinghouse locations and on a time sharing basis. A description of this code will not be presented here because of space limitations and because the primary intent of this paper is to demonstrate its application to long range planning.

4. Applications

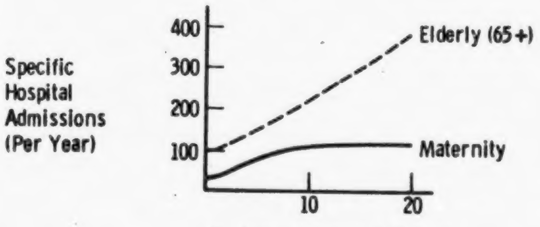
The purpose of this section of the paper is to briefly demonstrate how this simulation model has been applied so that the reader might perceive of ways of applying the methodology



(a) Base Year Population By Age, and Sex (1000's)

(b) Population Difference From Base Year by Age and Sex (1000's)

(c) Population Difference From 10 Year Period by Age and Sex (1000's)



(d) Planning Period (Years)

Fig. 6—Health care planning example

presented here to his particular problems. The examples demonstrated (Figs. 6 through 10) have been simplified to avoid the details that were obtained in thorough studies of these problems.

Applications are discussed in three areas: 1) Health Systems Planning; 2) Educational Planning; and 3) Urban Planning.

4.1 Health Systems Planning

The population characteristics projected for the planning period when used with the known incidence of disease for race, sex or income group provide a future demand forecast for health care facilities. This information is then used in determining the size and location of new facilities or the modification of existing facilities.

Consider the problem of forecasting hospital admissions for elderly and maternity care for a 20-year planning period as shown in Fig. 6. It is assumed that this hospital is servicing a rapidly growing suburban area in which developers are building 400 or 500 units per year over the first five to eight years of the planning period. From the eighth to the twentieth year moderate or little developer activity is assumed and the area grows at its natural rate (based on assumed fertility and mortality rates).

Results from the demographic model simulation are shown in Fig. 6. In (l) a graph depicting the population composition in the base year is shown. In (b) incremental changes from the base year are shown and (c) incremental changes from the 10-year projection are shown. Hospital admissions for both elderly (65+, 100 per 1000 pop. per year) and maternity (assuming 75 births/1000 females, ages 15-44) are plotted versus the planning period.

At least two areas of significance appear from an examination of the results. First, the planner should be cautious in overstaffing or building for maternity admissions early in the planning period and secondly, one must not delay too long in planning for elderly care to avoid a crisis situation late in the planning period.

This demonstrates just one of the many population related problems in health care planning that might be examined through simulation.

4.2 Educational System Planning

The projection of scholastics for the planning period is of direct value to educational planners in determining future enrollments which dictate the location, size, staff and material requirements for educational facilities. The projections may also be used indirectly to determine the financial resources or size of bond issue necessary for future facility construction and operation. In addition they can provide information which shows such situations as a peak in grade or middle school enrollments followed by a sharp decline. Situations like this may favor portable modules for schools rather than permanent construction.

Consider the problem of forecasting educational costs and resulting school tax burdens in a rapidly growing suburban area as defined in the previous section. Results obtained directly from a demographic simulation for this problem are shown in Figure 7. In (a), (b), and (c), the

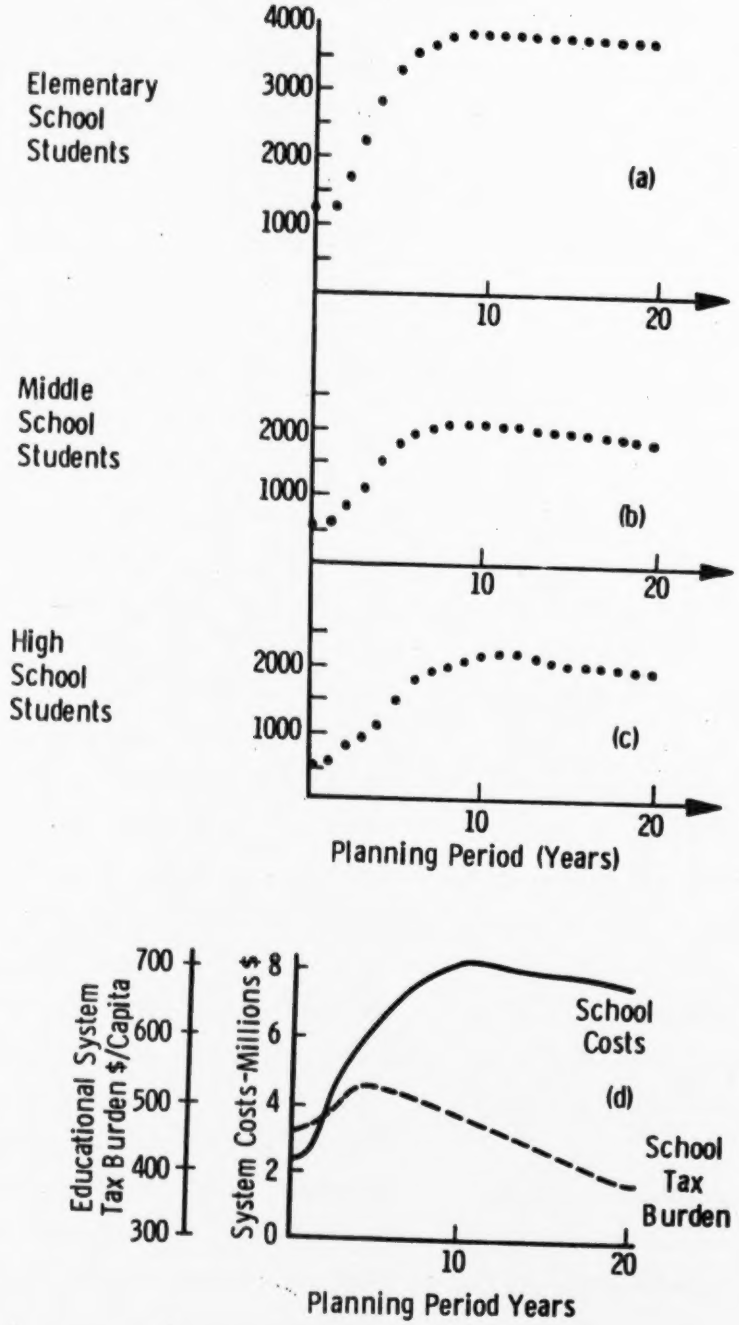


Fig. 7—Educational planning example

elementary, middle and high school students, as obtained from the demographic code, are plotted versus the planning period. In (d) educational system costs (assuming $1000 per year per student) and resulting tax burden are plotted for the planning period. The tax burden is approximated by dividing the educational costs by the sum of the cohort populations age 25 and above. In an educational system based entirely on real estate taxes this may be slightly inaccurate with regard to exact numbers but the trends as shown in Fig. 9 should still be valid. The inflation factor has also not been explicitly accounted for in the dollar values. It has been assumed that inflation has equal impact on the tax burden and school costs.

Examination of Fig. 7d shows that the tax burden rises during the development period (1-8 year planning period) but once the community has matured the burden actually falls below initial levels. This dynamic change in tax burden demonstrates the importance of a well planned development schedule.

4.3 Urban Planning

This digital computer demographic simulation should be helpful in investigating many urban problems. Some particular applications might include: 1) showing the effects of eliminating one type of dwelling unit (i.e. single family housing) and replacing them with another type (i.e. low income apartments), 2) effects of zoning law changes, 3) population shifting impact of new towns on existing communities, 4) transportation planning, 5) recreational facilities planning, and 6) low cost housing programs.

Some of the results from applying this simulation model to a study of the population and school enrollment for the Alief Independent School District, Harris County, Texas, are shown in Figs. 8, 9 and 10.

For this study the Alief area was divided into five regions as depicted by the simple map shown in Fig. 10. These regions are explicitly defined by the following.

Region 1 - north boundary, Alief Independent School District; west boundary, Barker Reservoir; south boundary, Fort Bend-Harris County line; east boundary, feeder to Katy Highway.

Region 2 - north boundary, Alief Independent School District; west boundary, feeder to Katy Highway; south boundary, Fort Bend-Harris County line; east boundary, Synnot Road.

Region 3 - north boundary, Alief Independent School District; west boundary, Synnot Road; south boundary, Alief Jeanetta Road; east boundary, Alief Independent School District.

Region 4 - north boundary, Alief Jeanetta Road; west boundary, Synnot Road; south boundary, Bissonnet Road; east boundary, Alief Independent School District.

Region 5 - north boundary, Bissonnet Road; west boundary, Synnot Road; south boundary, Fort Bend-Harris County line; east boundary, Alief Independent School District.

Figure 8 shows a high, average and low population projection corresponding to three different sets of input data to the demographic code for a planning period of 12 years beginning in 1968 and continuing through 1980. Some of the necessary input information for computing the saturation population of the area is shown in Figure 9.

Figure 10 shows the distribution of the population which was constructed from the simulation output for two years, 1975 and 1980. This output has been used by Alief planners in estimating future community needs.

6. Conclusions

The development, application and simulation of a demographic model suitable for long range planning has been described.

Applications of the simulation model to problems in health care, educational and urban systems planning have been presented.

7. References

1. U.S. Bureau of the Census, "Census Use Study: General Description," Report No. 1, Washington, D.C., 1970.
2. U.S. Bureau of the Census, "Americans at Mid Decade," Series P-23, No. 16, Jan. 1966.
3. Department of Administration, Bureau of State Planning, "Wisconsin Population Projections," distributed by Document Sales, State Office Building, Madison, Wisconsin, April 1969.
4. Keyfitz, Nathan, Introduction to the Mathematics of Population, Addison-Wiley, Reading, Massachusetts, 1968.
5. Benrud, C. H., "Systematic Procedures for Population Estimates and Projections for North Carolina," Vols. 1 and 2, Research Triangle Institute, N.C., PB 190 224, February 1969.
6. Benrud, C. H., "A Review of Population Estimation and Projection Procedures with Special Reference to Small Areas," Working Paper No. 1, Project SY-388, Research Triangle Institute, December 1968.
7. Lowry, I. S., "A Model of Metropolis," Rand Corporation Memorandum RM-4035-RC, August 1964.

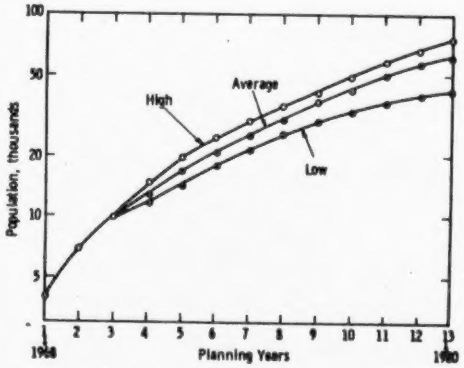


Fig. 8—Alief Area, Harris County, Texas. Population Projection 1968-1980

Demographic Region	Percent of Area Available For Residential Development
1	60
2	50
3	20
4	60
5	60

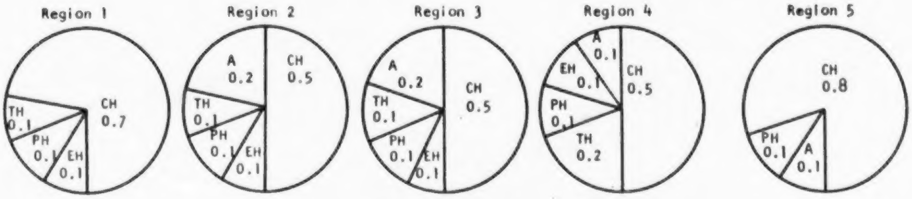


Fig. 9—Development input data for demographic modeling code

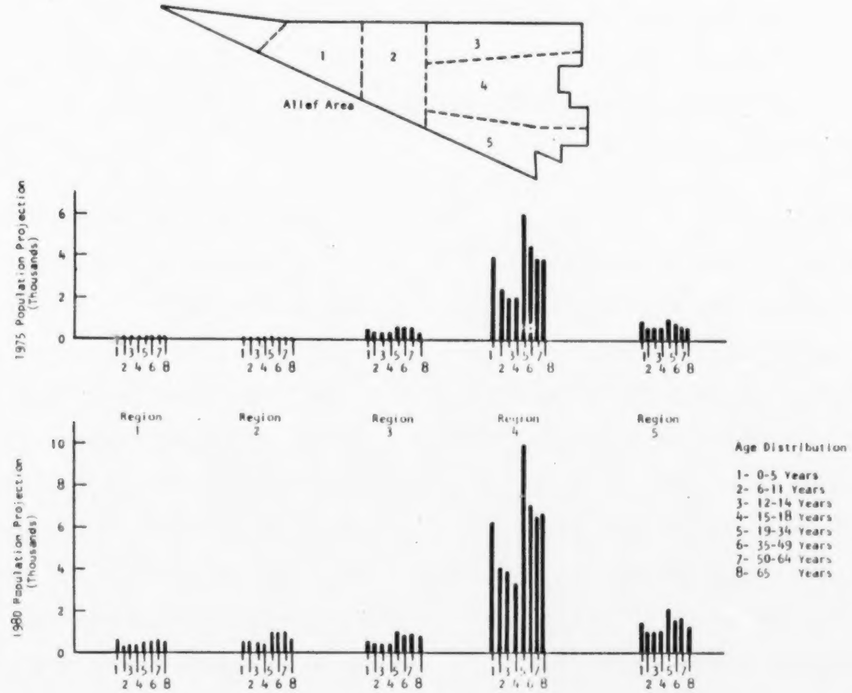


Fig. 10 —Demographic regions & population distribution for alief area, Harris County, Texas

SIMULATION OF NEW YORK CITY'S FELONY ADJUDICATORY SYSTEM

Lucius J. Riccio
Department of Industrial Engineering
Lehigh University

Abstract

A simulation model is described in this paper that was part of an analysis of New York City's Grand Juries and Supreme Courts. It has been used to test the relationship between additional court resources and the length of delay in the courts and the number of defendants in detention facilities.

Introduction

Like many other states, New York has been considering the implementation of court procedures generally known as "speedy-trial" rules. The rules contemplated in New York with respect to non-homicide felony cases are the following:

- A defendant shall be dismissed if his trial has not begun within 180 days of arrest, not including defendant-caused delay; and
- A defendant shall be released on his own recognizance if he is in custody and his trial has not begun within 90 days of arrest, not including defendant-caused delay.

The State Legislature, fearful that the courts could not possibly respond quickly enough to meet the demands such rules placed on them, opted for the rules advanced by the District Attorneys' Association. The rules were identical to those above with the exception that the milestone to be reached, rather than the beginning of trial, is the "ready for trial" declaration by the District Attorney. That is, the District Attorney must be ready to go to trial within 180 days of arrest not including defendant-caused delay.

The benefits of realizing speedy trials would be very great. The cause of justice is clearly vitiated by the lengthy delays presently characteristic of felony adjudication. The prosecution's case is weakened as delays drag out court proceedings; evidence can be lost, witnesses and victims may forget important facts or may die. On the other hand, oftentimes defendants will plead guilty simply to escape the detention facilities which in general are not very pleasant places (in Manhattan, the Detention Facility is appropriately called "The Tombs"). These two aspects of the problem do not cancel each other, but rather widen the possibilities for injustice.

Other aspects favoring speedy-trial legislation are that by insuring the speedy flow of defendants through the courts, detention populations would be minimized, and a greater degree of satisfaction would be imparted to other law enforcement agencies, such as Police, who have become increasingly critical of the courts.

New York City's Felony processing system can be simply described in the following way:

Felony arrests made by the Police are brought to the Criminal Court (Lower court)[1] for arraignment, at which time the defendant is informed of the charges against him. At this time, the case may be dismissed, the charge may be lowered to a misdemeanor, a plea may be offered[2] or the case may simply pass on to the next stage. If the defendant remains in the system he may be scheduled for Criminal Court appearances which generally lead to a hearing. However, many cases are taken out of the Criminal Court by the District Attorneys to avoid a hearing. In that way, they need not expose undercover agents. That occurs in many narcotics cases. Some defendants, for their own reasons, choose to waive the Criminal Court hearing.

The next step in the procedure is the presentation of the case to a Grand Jury. In addition to those cases described above, many cases start at this stage. District Attorneys often seek an indictment before they authorize an arrest. The Grand Jury can do one of the following things:

- Vote an indictment
- Dismiss the case
- Return the case to the Criminal Court for processing as a misdemeanor.

After an indictment is filed, the defendant is arraigned in the Supreme Court (Superior Court).[3] Pleas (which can be offered at almost any stage) can be offered here. That generally depends on the presiding judge in the arraignment part and whether or not the county has pre-trial conference (PTC) parts. The pre-trial conference parts, which are the next step in the process (where they exist) have recently been established to improve and institutionalize the plea-bargaining procedure. If the county does not have PTC parts, reappearances may be scheduled in the arraignment part to try to get a disposition without going to trial.

Following the arraignment and PTC parts are the trial parts.[4] Trial parts are designated as either Legal Aid, regular (private counsel), homicide, or narcotics. Usually a case will require a number of appearances in the trial part before the case is either disposed or made ready for trial. The length of adjournment between appearances is a function of many things, often a function of the cause of adjournment. For example, the failure of a witness to be present at the proceedings may cause an adjournment. The length of the adjournment will then be a function of the availability of that witness. Some other reasons for the variation in the length of adjournment are counsel (or the arresting officer) being on vacation and the availability of an opening on the court's calendar.

After all trial part proceedings have been completed and the defendant(s) has neither plead guilty nor had his case dismissed, the case is placed on the "ready and pass" queue. This queue contains all cases in the part ready to go to trial. A case gets placed in this queue when both parties declare readiness for trial or the judge decides there is no reason for further delay. A trial part will try only one case at a time.

Figure 1 is a diagram of the preceding description of the Felony Processing System. The status of the system at the beginning of this study can be summarized as the following:

As of the end of 1971, approximately 40% of all felony cases city-wide required more than 180 days from the time of arrest to disposition or first trial appearance. About 65% of the felony cases in detention need more than 90 days for disposition or first trial appearance. Detention populations in city prisons were in the 120-200% capacity range. The City was contemplating building a new "tombs" to accommodate the increased detention population. The cost of the facility would be in the neighborhood of $60 million, or about $48,000 a bed, with the benefits, at this time, to the criminal justice process at best being questionable.[5]

Planning for the Speedy-Trial Rules

Although the installation of the speedy-trial rules had been expected for nearly a year, it was not until the winter of 1972 that an analysis of Supreme Court needs was undertaken. At that time two were performed; one by the New York City Budget Bureau and the other by The Committee on Court Delay (an Ad Hoc Group formed from the major city and state agencies dealing with the courts). The two analyses were similar both in approach and in conclusions. The Budget Bureau recommended the funding of 35 new trial parts and 1 additional Grand Jury while The Committee on Court Delay suggested 30 new parts and 4.2 Grand Juries. Most of the recommended trial parts would be temporary; their purpose would be to help eliminate the backlog.[6] Judges would be "borrowed" from civil case processing to man the trial parts. The cost of the new parts and Grand Juries was set at 3.7 million by the Budget Bureau while The Committee on Court Delay projected a $12 million expenditure to fund their recommendations.

Both studies employed reasonably similar input-output techniques for each county, the number of defendants disposed of in 1971 was divided by the number of trial parts to obtain a measure of trial part productivity. They divided that figure into the total number of defendants presently awaiting a trial part appearance to arrive at the additional parts needed to eliminate the backlog. Also, the productivity measure was divided into the expected increase in cases that will reach trial parts in the coming year to find how many parts will be needed to handle the greater case load.[7] The sum of these two calculations was the recommendation for additional trial parts. The Grand Jury figures were arrived at in a similar fashion.

Certainly a massive infusion of resources will reduce the backlog and speed up processing times. However, it is questionable whether such a dramatic increase is required. The need for an additional phalanx of trial parts to reduce the present backlog may not be real. The need would be a function of the amount of delay caused by resource constraints. The number of cases pending will approximately equal the average number of cases arraigned per day times the average number of days for case disposition. The average number of days for case disposition would approximately be equal to the number of appearances times the average length of adjournment between appearances. With that in mind, it is clear that diminishing returns for additional trial parts may be reached rather quickly. The addition of parts will reduce the length of time between appearances. However, there are limits to that reduction because as discussed earlier, there are other reasons besides limited court time that affect the length of adjournment. Thus a point will be reached where the addition of more trial parts would only result in smaller calendar sizes and not a shortening in the time to disposition.

The trial part productivity measure is also somewhat misleading. Productivity, as it has been defined in the mentioned studies, is a function of many things, one being the size of the backlog. Productivity would increase as the backlog increased until the backlog is large enough to maintain full calendars. After that point, productivity would not increase as greatly and that increase would probably be the result of lower plea offerings.[8] Productivity is really a function of the number of defendant-appearances per day presided over by a Judge. Thus once calendars are full, and if all else remains the same, productivity can increase little.

The number of dispositions would also be quite sensitive to the amount of input. This is because although many cases take a long time for disposition, many cases "plead out" early in the process and also because as the backlog increases, greater pressure is placed on the system to dispose of cases. Essentially what I am saying is that "productivity" is a function of many things, and that additional parts may not be the entire answer.

Finally, the input-output models do not take into account the vicissitudes of some parameters. In the past, many parts would shut down during the summer and all parts would shut down for two weeks at Christmas. This is particularly debilitating in Manhattan.

The Committee on Court Delay recommended a list of improved court procedures such as standardized adjournment durations and calendaring procedures. If these recommendations could be implemented, it is quite possible only a few new parts (or maybe none at all) would be needed. However, they did not attempt to quantify and predict the effects of such improvements or whether they could even be implemented at all.

Neither study attempted to uncover the precise functional relationship between additional court resources and the amount of delay in the system. Neither could say that their recommendations will reduce the delay in an optimal fashion, optimal in accordance with some well-defined criterion. Clearly that must be the objective of any detailed study dealing with court delay and speedy trial rules.

The Simulation Model

Recognizing the deficiencies of those studies, The Mayor's Criminal Justice Coordinating Council (CJCC) and The Budget Bureau authorized the development of a computer simulation model of the court system to analyze specifically the problem of delay as it relates to the speedy trial rules.

The vast majority of delay is due to the wait between appearances in one processing element or another. The present amount of delay can be described by frequency distributions generated from sampling studies. Such sampling studies are snapshots of the system at a given point in time, but are not of great value in trying to predict how the delay will change with changes in system resources. Waiting time is a function of many things, one of which being the size of the backlog. Thus it was projected that a simulation model that incorporated all of the vital aspects of the system would be able to generate from within the delay distributions associated with various levels of resource allocation.

It was decided that the model would not simulate the Criminal Court proceedings, considering the time constraints on this study. The Criminal Court is replaced in the model by a probability distribution that describes the delay from arrest to held for Grand Jury.

The model is a descrete-event simulation updating itself on a daily basis.[9] All work is accomplished on weekdays. However, weekends are included since they count toward the speedy-trial rules. On each day it schedules new arrivals, "calls the calendars" of all grand juries and parts, and schedules trials. An "arrival" is a case placed in the held for Grand Jury queue. All of the processing units function in the following way:

When a case first arrives at a station, it is assigned a priori the number of appearances it will require for a disposition at that stage. It is then scheduled into its first appearance. The number of appearances remaining is retained and with each appearance, it is reduced by one. When all appearances have been completed, the model determines if the case goes on to the next processing stage or if it leaves the system.

Probably the most critical aspect of the model and the real system is the calendaring procedures. It is critical in that it is very difficult to ascertain precisely what is done. The philosophy behind the model's scheduling algorithms can be broken into two parts. The first places limits on the amount of cases that will be scheduled for a working day. An "opening" is considered to exist on a day if on that day fewer cases than the limit have been calendared. Some scheduling is done by assigning a case to the first available opening. First appearances are generally assigned in this way. The second aspect to scheduling recognizes the multiplicity of causes that affect the lengths of adjournment. As such, the length of adjournment is found with the aid of a probability distribution. The date chosen for the next appearance is checked to be sure that the limits mentioned above have not been violated. Most re-appearance scheduling is performed in this fashion. All processing units give priority to jail cases over bail cases and re-appearances over first appearances. All criminal justice officials queried agreed with this formulation.

The model earmarks some Grand Juries and trial parts for homicide cases. Those units may handle other cases if there are openings on their calendars. Trial parts dispose of their daily calendars and then, if a case is ready, schedule a trial.

Many simulation models have the built-in assumption of independence between stages and uniformity within stages. Because each defendant's vital data is stored by the program and "carried" from stage to stage, this model is not restricted by the assumptions of independence and uniformity. Many variables were tested for functional dependency and categorical peculiarity. Jail and bail cases were handled differently throughout the model. Also, homicide cases generally required different parameters than non-homicide cases. However, most variable-pairs were found to have virtually no discernable amount of interdependency; e.g., the number of appearances prior to trial part seemed to have no affect on the number of appearances in a trial part. Those variable-pairs found to be in some way interdependent will be mentioned later in the text.

Input to the Model

As part of their study, The Committee on Court Delay commissioned a sampling study of felony cases that had reached disposition in the year November, 1970 through October, 1971. That study provides a significant amount of reasonably good data. The following information was generated from their study.

As mentioned earlier the Criminal Court processing would be represented by a probability distribution. Figure 2 is a graph of the time from arrest to held for Grand Jury for all cases. The model used two distributions - one for jail cases and one for bacil cases, each having the same shape as Figure 2 but with averages of about 2 weeks and 7 weeks, respectively. The distribution in Figure 2 yields an average time of 4.73 weeks from arrest to held for Grand Jury with approximately 70% of the cases requiring one month or less to reach the held for Grand Jury stage. The distribution may have improved since the time that those cases sampled went through this segment of the system. However, this distribution is representative of system performance described as "current" which in court parlance is the proverbial goal of all court administrators. Current means the court is disposing of as many cases in a month as it receives and the time to disposition is within a certain specified criteria. Backlog can then be defined as the number of cases that cause a violation of the above constraints. The definition of backlog used by this author has been and will remain synonymous with the number of cases pending. This is consistent with the usage of the term by the other studies referenced.

The sample also provided some information on the number of appearances required for disposition at arraignment and in trial parts. Figure 3 is the frequency distribution of the number of appearances that were required prior to trial part consideration. Since there were no PTC parts in Manhattan, all of these appearances were in the arraignment part. Figure 4 is the frequency distribution of the number of appearances in trial part required for disposition or readiness for trial. Very little is known about how fixed these distributions are, whether they will change with changes to the system.

As stated earlier, the number of appearances required for disposition at a stage is assigned to a case when it first arrives at that stage. When all appearances have been completed, the model determines if the case goes on to the next stage or if it leaves the system. Since this model is not restricted by the assumption of independence, the possibility of going to trial was viewed as a function of the number of trial part appearances. This hypothesis was "tested" using the sample data. However, there were so few cases that went to trial (3.8% of the sample) that it was quite difficult to draw conclusions. Figure 5 shows both the fraction of cases going to trial for each number of appearances and the assumed probability distribution. The plot means that, reading on the assumed probability line, a case which required 10 appearances in a trial part has a 5% chance of going to trial when its tenth appearance has been completed.

Once a case has reached trial it is assigned a length of trial. Again, there is little data on this subject. Ms. Virginia Ambrozini, a consultant to CJCC, performed a study of Supreme Court operations in the summer of 1971. Figure 6 shows the results of her study with respect to the length of trial. Figure 6 also shows an assumed distribution of length of trial. This parameter could be quite sensitive to the speedy-trial rules. Presently, some defendants plead guilty as soon as the first juror is called. (That is the landmark signifying the beginning of trial.) They plead at that time knowing they have no case; they got as far as trial possibly hoping for a lower plea offering. Approximately 20% of the trials are disposed of on the day they were begun. With the implementation of speedy-trial rules, at least as they were originally constituted, this distribution might shift. More defendants might wait until trial before pleading. After the first juror is called there is less advantage to continuing.

A very important aspect of the model is that part dealing with the length of adjournment. This is certainly an under-researched area. However, the Ad Hoc Committee's sampling study does provide some information regarding that parameter. In that study, for each case reaching the trial part stage, they recorded the date of first trial part appearance and the date of the commencement of trial, plea, or dismissal. The difference between these two dates divided by the number of appearances minus one is the average time between appearances. Figure 7 is a plot of the frequency of those average times. This distribution does unfortunately include those cases that did go to trial. Those cases include the time between their last trial appearance and commencement of trial in their averages. That could account for the four data points having over 100 days between appearances (which I considered to be outliers). This plot was used as an approximation for the lengths of adjournment.

The length of adjournment distribution is reflective of all the causes of adjournment and the causes of variation in the length of adjournment. Imbedded in it is, of course, the resource restrictions placed on the calendaring procedures. To run the model with accuracy the part of the distribution caused by resource restrictions and manifesting itself in terms of calendaring interference should be subtracted out. This is important because the lengths of adjournment will fluctuate with the level of resources. The way the subtraction will be accomplished will be by running the model with a fixed calendaring procedure and comparing the distribution obtained with Figure 7. Admittedly, this is a crude technique.

It was thought that there was a possibility that the average time between appearances might increase as the number of trial part appearances increases, the reasoning being that the same reasons that cause many appearances might also cause longer adjournments. Figure 8 is a graph of the average time between appearances versus the number of trial part appearances. It appears that there is a slight upward trend in the averages. However, note the rapid drop in observations just as the chart tends to move upward. Since the number of observations was small, it was assumed that the number of appearances has no effect on the length of adjournment. It was important to consider this point because if there was a correlation either positive or negative, it would have an impact on the spread of the distribution of time through the trial part phase of the system.

The other aspect of calendaring discussed earlier is the maximum number of defendants a trial part will schedule for a day. Since all parts give priority to jail cases over bail cases and re-appearances over first appearances, the model uses two limits for scheduling. The lower limit applies to first appearances for bail cases and the upper limit applies to jail cases and re-appearances. Preference is also shown by attempting to schedule jail case appearances with shorter average lengths of adjournment. Unfortunately, although this is an agreeable formulation, there is little data that provides an accurate assessment of those limits. Figure 9 is a frequency distribution of trial part calendar sizes obtained from a sample, compiled by the author, taken from the listings of trial part calendars printed in The New York Law Journal. It is a very small sample; however, a pattern is clear. A further study would probably show that the calendar size of a trial part is a function of the judge. From Figure 9 I selected 15 as the lower limit and 20 as the upper limit for all non-homicide trial parts. Of course, judges get holidays and sick days and as a result judges sit on the average only 4 out of every 5 weekdays. The model compensates for the sick days and holidays by lowering the limits by 20% to 12 and 16, respectively.

Very little of the data needed for the model is known about Grand Jury presentations. There are no statistics available concerning the number of appearances or the length of time between appearances. It is assumed, before more information can be collected, that the distribution of appearances drop off sharply after one, much like arraignment.

When building the model, the question of work units arose. What constitutes a unit of work for a trial part? For a Grand Jury? It would seem a case would be the standard unit of measurement of judicial performance. However, virtually all court statistics are presently based on the number of defendants that pass through the various processing elements. Records are kept by defendant-count for several reasons. One is that detention facilities house defendants, not cases. Another is that many defendants may be charged under one indictment or many indictments can be charged to one defendant. As the defendant traverses the system some indictments may be dropped, others consolidated. The ratio of defendants to cases will change several times during processing in a way presently unknown. Therefore, to bypass the difficulty of the defendant to caseload conversion, the model uses defendants as the operational unit of measure.

The model calculates a figure for the number of defendants in custody awaiting Grand Jury or Supreme Court appearances. Each defendant is assigned either a jail or bail status at the held for Grand Jury stage. A remand rate of about 75% is used in the jail-bail decision. (Of course, few defendants are actually remanded. Defendants in jail are for the most part people who cannot make bail.) The remand rate of 75% is held constant in the model not because in reality it actually remains constant but because it is subject to many forces and the function that governs the remand rate is not presently precisely known. The model is completely capable of incorporating a remand rate function and one should be included when a reasonable formulation has been arrived at. One factor to consider is the percent occupancy of the jails. It may be the case that when detention populations reach the 150-200% capacity range, judges give lower bail decisions.

Validating the Model

Before obtaining projections of future needs, the model had to be "validated". Since total validity is virtually impossible, a better description of this procedure would be "building confidence" in the model. There are two ways this is done. The first is by having confidence in the structure and input to the model. The second is by demonstrating that operating results for a controlled run reflect past experience.

The structural assumptions of this model have been continuously tested in discussions with Criminal Justice officials from many parts of the system. Most of the input to the model was acquired from The Committee on Court Delay's sampling study, a sampling of statistical repute.

Once the model was operational, it was "fine-tuned" to adjust for errors in assumptions and input for those situations where there was not enough data to provide accurate estimation. A test run was made of the year 1971. Figure 10 compares statistics obtained from the model against actual data. The distributions of times to disposition were quite similar in form to those obtained from the Court Delay Committee's sampling study, however with slightly smaller variances. The spread of the model's distributions was less for a number of reasons. Some of the inputed data is based on averages, the use of which tends to tighten rather than spread distributions. Judge productivity and calendar size were determined by an averaging process. The percentage of defendants pleading out at arraignment is a function of the judge sitting in the arraignment part. This could fluctuate much more than the model permits. The model is capable of incorporating these more accurate aspects and should include them as better data is generated.

Figure 10 demonstrates the fluctuations in the number of defendants awaiting trial part appearances during the twelve months covered by the committee's sampling study. Also on that graph is the simulation's results for the test run.

In the Fall of 1971 Manhattan added a pre-trial conference part. All cases go through this part between arraignment and trial part. Since there is little data about this part, it was assumed that almost all cases now have only one appearance in the arraignment part and the remaining appearances prior to trial part in the PTC part. That is the total number of appearances prior to trial part is still the same as before the addition of the PTC part. The capacity of the PTC part was established, in much the same way as the trial part limits, at 30 defendants per day. In February 1972, two trial parts, designated specifically for narcotics cases, were added. In the Spring of 1972, a grand jury was added. This last addition accounted, quite naturally, for a rise in indictments and a dramatic reduction in the number of defendants awaiting grand jury action. This met with widespread approval. However, the hidden effect of adding a grand jury was to send a sudden jolt to the rest of the system. Rather than reducing the total number of defendants in the system, it simply shifted the burden to the trial parts. The sudden rise in defendants awaiting a trial part appearance is really a transient effect rather than a steady state condition. Table 1 shows the total number of defendants in the system at three points in time. Admittedly, this is not a complete analysis; however, I think the point is clear.

	Oct 30, 1970	Oct 30, 1971	May 18, 1972
Grand Jury	530	523	298
Arraignment	682	530	322
Trial Part	2165	2198	2527
TOTAL	3377	3251	3147

Table 1

The behavior indicated above was demonstrated in a test run of the first half of 1972. That run and all others following included a 20% increase in grand jury presentations over 1971.

Using the Model

As discussed earlier the purpose of the model is to determine what is required to reduce processing delays, the backlog of cases, and the detention population. The model was run simulating 400 days beginning with September 1972 using the system conditions monitored at that time.

The model was run without any further additions of resources other than those mentioned in the last section. By September the jolt given the system in the Spring by the additional grand jury had worn off. The model projected for the months September through the first half of December 1972 a decrease in the number of defendants awaiting trial part appearances from about 3200 to 2600. The two-week shutdown in December would boost the backing log up to nearly 2800 but during the first six months of 1973 the model projects a steady decline to about 2300, at which time the total number of defendants in the system would be just under 3000 and the detention population would be about 1670.

Although the decline in the number of defendants looks encouraging, the percentage of cases taking less than 180 days for disposition is down only to 32%. Further analysis of the results indicated two things: (1) There would be an apparent bottleneck in the PTC part; and (2) the addition of trial parts would do little towards the reduction of delay unless tighter controls were placed on the lengths of adjournment. The average length of an adjournment in the trial parts was still over 18 days and the average calendar size in a trial part was under six defendants per day.

The model was run, starting again in September, with the addition of one PTC part. This expedited case processing dramatically and reduced the backlog to about 1725 defendants and a detention population of 1150 by June of 1973. However, the percentage of cases requiring more than 180 days for disposition was still high at 26%. The reduction of that percentage from 32% to 26% was almost entirely the result of the ability of the additional PTC part to get defendants to plead out before going on to a trial part.

The addition of the PTC part shifted some of the burden to the trial parts. However, the average calendar size per trial part went up only to 6.3 defendants per day. Thus it is clear that it is not more parts but tighter controls on adjournment lengths that are needed for a reduction in delay. To compensate for the low calendar sizes two additions to the model were made: (1) A calendar size control mechanism, and (2) an emergency scheduling algorithm for cases approaching the 180 day limit. They are explained below.

The system to a certain extent is self-regulating with respect to adjournment lengths. As the backlog drops the average length of adjournment should drop. A simple control mechanism was incorporated whose purpose was to adjust adjournment lengths so that average calendar size would not drop much less than 6. Six was chosen because previous runs of the model showed six to be a consistent figure for that parameter, and assuming new trial parts will function, at worst, like old ones, six seemed reasonable. A lower limit was placed on adjournment lengths to account for uncontrolled variables.

No matter what preferences judges or DA's have for size of workload, it is certain that no one will want to be given blame for allowing someone to "escape" under the 180 day rule. Thus, as a case approaches that mark, both will accept slightly larger calendar sizes resulting from shorter adjournment lengths. To reflect this, the model was set-up so that when a defendant was in the system for more than 140 days he would be allowed adjournment lengths of 5 days on the average.

The model was run with these additions, again using 2 PTC parts and 17 trial parts. In that run, average calendar size rose only to 6.6 defendants per day. The number of defendants awaiting trial part appearances plummetted to less than 1500 in the first 4 months of the simulation. However, the percentage of defendants requiring more than 180 days for disposition dropped only to 22%. Thus it was clear that concentration solely on the trial parts was insufficient because many cases were coming to their first trial part appearance having already logged in nearly 180 days. Much of the pre-trial part delay was, from the model's point of view, caused by excessive delay of bail cases in the criminal court. This was the only leverage point left in the system.

As stated earlier the distributions used by the model for criminal court delay were probably somewhat outdated due to improved administration within that court. The model was run as described in the paragraph above with all cases using the distribution previously used for jail cases only. That is, the model was run using an average criminal court delay of 2 weeks for all cases. It is quite possible that in reality this reduction has already been realized or it could be realized without an addition of resources.

The results of that run show only 18% of defendants requiring more than 180 days for disposition. This figure is an adequate system goal from the point of view of the model for four reasons. First, not all delay is caused by the state; some is caused by the defendant (request of an additional appearance or extra long adjournment) and as such is not chargeable to the 180 day rule. Second, all indicators point to the fact that the system has enough slack to allow additional special expediting of cases near the 180 day limit. Third, the 18% figure includes homicide cases (most of which take more than 180 days) which are not covered by the rule. Fourth, the 18% figure also includes some cases arrested before the rules went into effect. That is, the true figure for cases covered by the rule is probably between 5-12%.[11] Taking into account defendant caused delay and possibilities for additional expediting, the system resources defined in this last computer run will be adequate to meet the requirements of the rule.

Conclusions

The results obtained from the model point to the following recommendations for Manhattan:

1. Add 1 PTC part.
2. Add trial part emergency expediting mechanism for defendants in the system for more than 120 days.
3. Reduce Criminal Court delay to an average of 14 days for felony cases.

Many aspects of the model could change, such as the number of appearances required for disposition, and as such would change the results and recommendations herein described. However, these predictions are based on rather conservative data. It seems unlikely, with the low utilization rates of most resources and the urgency of the situation, that any aspect of the system will relax and cause the model's results to be overly optimistic. The only parameter change that could possibly cause major repercussions would be the number of grand jury presentations. However, even a large increase in that variable would not cause great damage since the constraining relationships in the system are not due to lack of resources.

This model has demonstrated its usefulness in the analysis of this court "crisis". However, it can, and should be, used to test system sensitivity to various parameters and to study other aspects of court system behavior. This simulation model is a powerful tool, useful to the process of understanding and improving court operations.

Notes

1. The Criminal Court disposes of all misdemeanor and violation cases in addition to being the first stage of the Felony Processing System.

2. Plea Bargaining is a means of disposing of cases before they tie up valuable court resources. The vast majority of defendants adjudged guilty are disposed of via a guilty plea. The defendant is induced to plead guilty by being offered a shorter sentence than the expected sentence if found guilty as the result of a trial. The plea offering may be a plea of guilty to a lower charge.

3. The Supreme Court handles only

felony cases.

4. A trial part like an arraignment or PTC part is the term used for a courtroom staffed by a Judge, District Attorneys, and clerical staff.

5. In addition to the original cost there would be about $12 million in annual operating cost. Also, it will take a considerable amount of time to build the jail.

6. There are many definitions of the term backlog used to describe system status. The definition used in this paper is the number of cases pending.

7. Budget Analysts were more perceptive in this calculation. They took into account the fact that the present number of parts was insufficient to handle properly last year's caseload. Thus, before considering the expected increase in caseload, they added parts to allow the court to be a match for last year's input rate.

8. The backlog could be eliminated quickly if the court decided to lower significantly its plea offerings. That fact leads to difficulties in predicting system behavior; but more importantly it shows the imprecise nature of felony adjudication. Lower plea offerings would have the effect of lowering the average number of appearances for disposition.

9. New York City has five counties each being a Judicial District and each has their own Criminal and Supreme Court. The model is set-up to simulate the operations of each county separately. The majority of quantitative information in this paper refers to Manhattan.

10. See Jennings, "Quantitative Models of Criminal Courts," 39th National Meeting of ORSA, May 5, 1971.

11. The model was set-up so that those cases not affected by the rule would not have statistics collected about them. That run showed only 2.8% of defendants took more than 180 days for disposition. Another run was made setting the maximum trial part calendar size at 10 (had been 16). In that run 12% of defendants affected by the rule took more than 180 days. The backlog was reduced to a projected figure of 1700 defendants by June 1973.

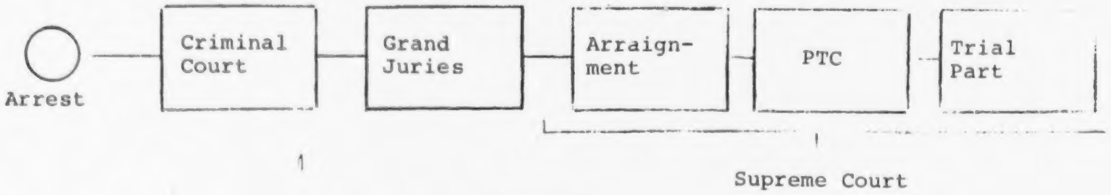

Arrest
Criminal Court
Grand Juries
Arraign- ment
PTC
Trial Part
Supreme Court

Figure 1
Felony Processing System

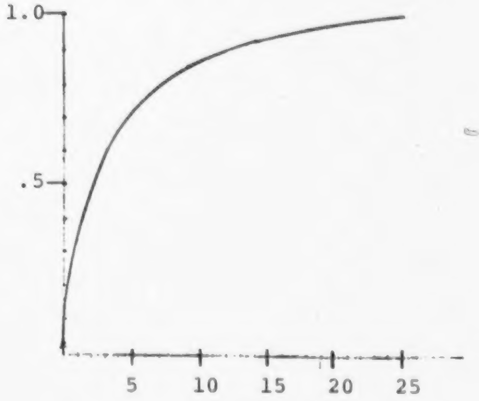


Figure 2
Criminal Court Delay (Weeks)
Cumulative Distribution

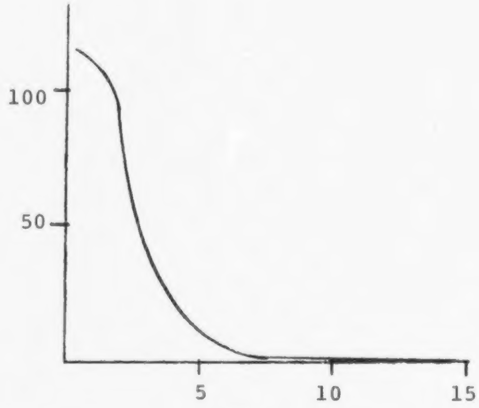


Figure 3
No. of Appearances Prior to Trial
Part Frequency Distribution

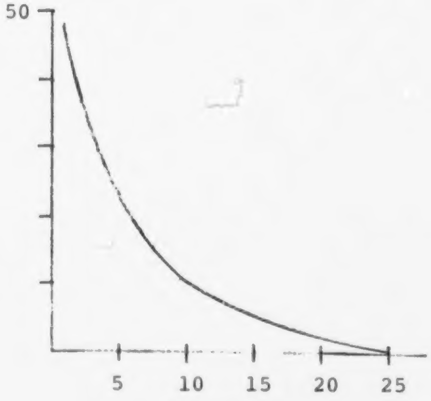


Figure 4
No. of Appearances in Trial Part
Frequency Distribution

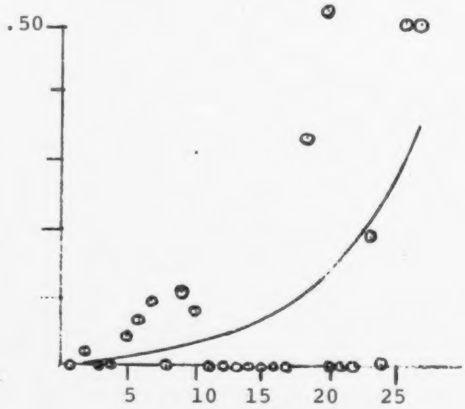


Figure 5
Probability of Trial vs. No.
of Trial Part Appearances

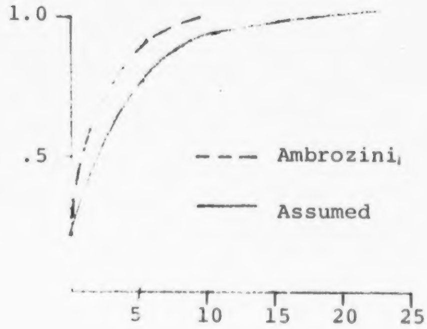


Figure 6

Length of Trial (Days)
Cumulative Distribution

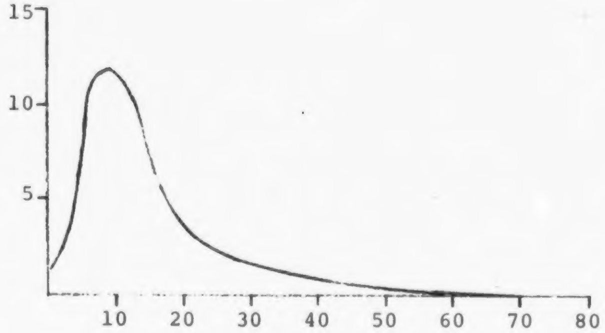


Figure 7

Length of Adjournment (Days)
Frequency Distribution

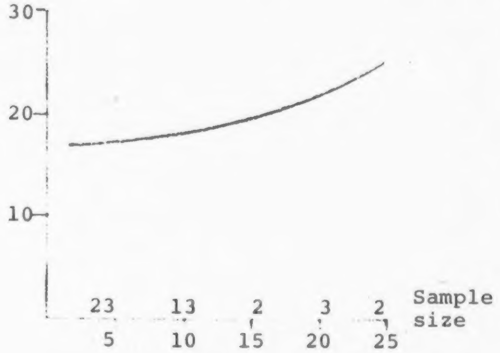


Figure 8

Length of Adjournment (Days)
vs. No. of Trial Part Appearances

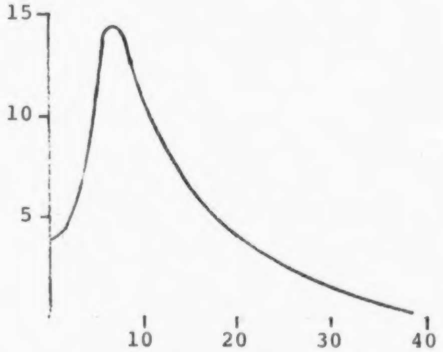


Figure 9

Trial Part Calendar Size (Defend.)
Frequency Distribution

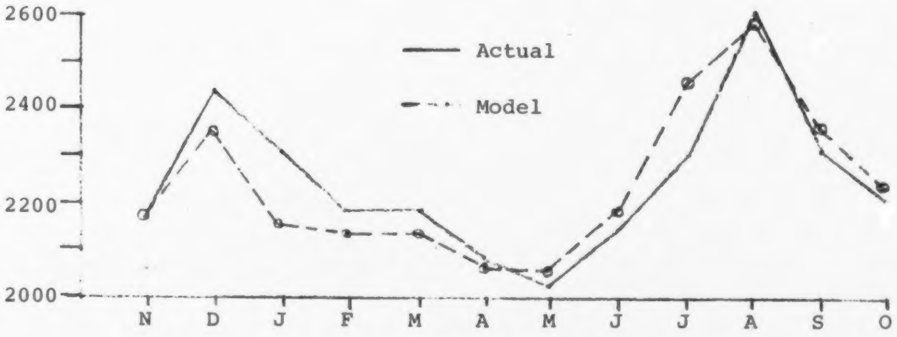


Figure 10

Actual vs. Model - Nov. '70 - Oct. '71

A SIMULATION MODEL OF THE NEW YORK CITY FIRE DEPARTMENT: ITS USE IN DEPLOYMENT ANALYSIS

Grace Carter, Edward Ignall,* Warren Walker*

The New York City-Rand Institute

Abstract

This paper describes a simulation model developed as a tool to aid deployment decision-making for the New York City Fire Department. The model, written in Simscript I.5, has been used to evaluate alternative solutions to workload and response problems plaguing the City. These solutions involve new policies for locating, relocating and dispatching fire-fighting units to achieve a more effective utilization of resources. Several specific applications of the simulation are described. In addition, methodological issues concerning the design and use of the model are addressed.

I. INTRODUCTION

Since 1968 a large-scale research program for the Fire Department of the City of New York has been in progress at The New York City-Rand Institute. An overview of that research is contained in [1]. A major part of the program has been an investigation and evaluation of alternative policies for the deployment of fire-fighting resources. This paper describes a simulation model of the responses of fire-fighting units to alarms, which has been one of the tools used for this evaluation. A comprehensive description of the design of the simulation model is given in [2]. Our emphasis in this paper will be on the use of the model. We will describe the various policies which were compared and present the differences in their performance. We will also note new policies which have been implemented based in part on simulation results.

---

*Also Columbia University.

We had two primary motives for using simulation:

(1) To be able to compare alternative policies without risking lives and property and spending considerable sums of money by trying them in the real world; and

(2) To gain a better understanding of fire department operations by making clear the effects of interactions within the system and the second-order consequences of suggested actions.

The need for such understanding and for the development of new policies had become apparent to the Department as the alarm rate began to grow exponentially in the 1960's. In 1968 the Department responded to 227,000 alarms, more than three times the number responded to in 1956, placing a severe strain on existing fire-fighting resources and increasing the pressures on the City for adding to them. However, a full-time fire company--a pumping engine or ladder truck and a complement of men sufficient to man it around the clock--costs over $600,000 per year. The Fire Department wanted to make more efficient use of existing resources and to determine the most effective ways to add new resources. The simulation became the major tool for evaluating candidate policies.

We present here the results of simulation experiments in which three types of policies were tested:

(1) Dispatching policies: how many and which units of each type (engines and ladders) should be sent to each incident? With some exceptions, in the past the same response was sent to all alarms received by street box in all areas during all times of day. However, box histories showed that the probability that a given alarm signalled a serious fire varied greatly by time and area. With the simulation it was possible to observe the effect of varying the standard response.

(2) Relocation policies: given several units busy in an area (at one large fire or several small ones), which available units should be moved into empty fire houses in the area, when should they be moved, and which empty houses should they fill? The method being used by the Department was not designed to handle the increasingly common instances of several fires in progress in an area simultaneously.

(3) Allocation policies: how many units of each type (engines and ladders) should be located in each area at each time of day, and where should their houses be? Traditional practice in New York City and throughout the country is to have the same number of units on duty 24 hours a day, although, for example, the alarm rate in New York City between 8 p.m. and 9 p.m. is six times as high as between 5 a.m. and 6 a.m., and over 60 percent of the day's alarms are received between 3 p.m. and midnight.

The new dispatching, relocation, and allocation policies tested in the simulation were designed on the basis of the experience of fire officers and analysis by officers and members of the Institute's staff. The purpose of the simulation experiments was to compare the new policies to the traditional ones under controlled conditions to see how much better (or worse) the new ones would be expected to be.

II. MEASURES OF EFFECTIVENESS

A direct way to measure fire department performance is in terms of loss of life and property damage, but we made an early decision in the design of the simulation model not to attempt to do so. The relationships between deployment actions and these direct measures were not known, and obtaining such measures seemed to be a long and possibly futile project. We felt that our immediate objectives could be met by using several easily calculated surrogate measures of performance such as the response time of units to fires. It developed that different policies could often be ranked in terms of performance in preventing life loss and property damage by examining their performance on the surrogate measures if one assumed only a monotone relationship between them and the direct measures.

By a surrogate or "internal" measure we mean an aspect of the Fire Department's performance which can be observed by watching only the Department's activity and not its consequences. Examples of internal measures are the number of units responding to incidents, the time it takes each one to arrive, the number of responses made by each unit, and the proportion of time each unit spends working.

In order to show the close interrelationship between direct and internal measures let us focus on one direct measure: loss of life. How would it be affected if the current policy were changed? In order to find this out from the simulation we would have to know, first, how the pattern of responses would differ under the new policy; and second, how this changed pattern would affect loss of life. The latter question is quite difficult to answer. On the other hand, the answer to the former is easy to obtain in a simulation, and this internal measure can be used as a surrogate for loss of life and loss of property. The following analysis shows how this may be done.

A useful measure of a policy at a particular incident is the vector of the response times of all units responding to it. The vector of response times for engines gives the time of arrival (relative to the time of alarm) of the first engine, second engine, etc. We divide the vector into two parts depending on whether the unit is an engine company or a ladder company. The vector for each type of unit is then put in order of unit arrival.

If, at a particular incident, one policy produces a response time vector every component of which is smaller than the corresponding component for another policy, and there are no other differences between the two policies, then it is clear that the former policy is as good or better for preventing loss of life, even though we do not know the precise relationship between response time and loss of life. We would not expect one policy to be better all the time, but we can aggregate the response time vectors by incident type and estimate the distribution of the response time vector for each type. (See Table 1 for one possible breakdown of incidents by type.) The empirical cumulative distribution function (cdf) for a given incident type can then be used to test whether one policy is better than another policy for response to that type of incident. If the grouping we are using for this analysis is those incidents at which arrival time is the primary factor in determining if a life will be lost, then, if the empirical cdf for policy 1 is better than that of policy 2, policy 1 is the better policy; that is, it will result in as few or fewer lives being lost.

Determining whether the empirical cdf under policy 1 is better than that under policy 2 is generally not an easy task. For example, policy 1 may get the first ladder to incidents faster while making the second ladder response slower than policy 2. In practice we have generally focused on individual components of the vector of response times. For example, we have compared different policies with respect to

- o the distribution of first ladder (engine, arriving unit) response times to different types of incidents
- o the average first ladder (engine, arriving unit) response times to different types of incidents.

We use standard statistical tests to determine if differences in these response time measures between policies are significant.

Other internal measures of interest are coverage (the current proportion of alarm boxes for which nearby units are available) and workload (the number of responses made by firefighting units). They are discussed in detail in Section IV.

III. THE SIMULATION MODEL

The simulation package consists of three parts:

1. An input program. Given a probabilistic description of the fire demands in the area to be simulated, this program, written in SIMSCRIPT I.5, generates the set of incidents (exogenous events) which will occur in the simulation. The description consists of:

   - o potential incident locations (we used alarm box locations);
   - o for each alarm box location, the arrival rate of each type of incident and the proportion of each type which are reported by telephone;
   - o a description of each type of incident, which includes the number and kind of companies required to handle the incident and the length of time each company is required.

2. The simulation program. This program, also written in SIMSCRIPT I.5, simulates the Department's response to a given set of incidents under a particular set of deployment policies. The program produces statistics on company utilization, workloads, response times, and coverage. It also produces output files that are available for later analysis. Usually, this program is run several times with the same input file but using different deployment policies.

3. Post-simulation analysis program. These are programs, usually written in FORTRAN, which measure the statistical reliability of the simulation output and make comparisons between simulation runs based on various measures.

The run time of the simulation program increases linearly with both the number of alarms processed and the number of times the simulation is interrupted to obtain samples of the state vector. Approximately 78 samples or 10.6 alarms can be processed in one CPU second on the IBM 360/65. A typical run might contain 3,000 alarms and 500 samples and thus require 288 CPU seconds.

The simulation was designed to facilitate the process of making policy changes. To write a program which would be able to simulate many different policies without being sure in advance what these policies would be required the use of interchangeable sub-routines and a flexible data base. We present below a brief summary of the organization of the simulation program and a description of the data base which was used in our experiments.

Simulation Flow

The progress of an incident can be traced by following the series of event routines through which it passes. (In what follows the names of event routines will be written in capital letters.) The FIRE first breaks out and, some time later, at an instant predetermined on the input tape, the ALARM is turned in. The program--using a given dispatch policy--decides which companies to send to the alarm and schedules a DISP (dispatch) event to occur after a one-minute delay to allow for the alarm to be processed at the dispatching office and for the men to climb onto the apparatus. In the DISP event an arrival event (FARV) is scheduled for each of the dispatched units. The arrival time depends on the distance between the fire and the unit. The responding (and returning) units are assumed to travel at 20 mph, and a combination of right-angle and Euclidean distances is used to determine response distance.

The first of the FARV events to occur for a particular incident produces a CALIN (Call In) at which the first arriving unit "reports" the condition of the incident to the dispatching office. If too many units have been sent some are directed to return to their houses; that is, the FARV events of the excess companies are cancelled and HARV (House-arrival) events are scheduled for them. During their return home these companies are available for dispatch to other alarms. If the fire is a greater alarm fire, HARLM (higher alarm) events are scheduled, resulting in more DISP and FARV events until enough equipment is at the scene of the fire. The RELS (release from service) events are scheduled at times which depend on the arrival time of the company and the work time parameters found on the input tape. After release, companies proceed back to their fire stations, causing HARV events.

The details of the event routines and the output statistics have been tailored to New York City. However, metropolitan fire operations are sufficiently similar across the country that the basic structure of the simulation should be applicable to other cities.

Data Base

The Bronx, one of New York City's five boroughs, was the subject of the simulation experiments reported on here. Seven incident types were defined based on the number of units required and the amount of time each unit would work. The incidents ranged from false alarms, which require only a short search by an engine and a ladder to assure that there is no fire, to third alarm fires, at which fifteen companies work for several hours each. For each type of incident work times were treated as constants, made equal to our estimates of mean work-time. The simulation also distinguishes between alarms received by telephone and those turned in from street boxes, since most dispatching policies depend on how the alarm is received. Table 1 lists the seven alarm types together with the number of units each requires and the other incident characteristics.

To reduce computer storage requirements, the 2,500 alarm boxes in the Bronx (roughly one on every second street corner) were gathered into 358 relatively homogeneous box groups. Each of these box groups is then simulated as if it were a single alarm box. The location of each group was defined as the centroid of the boxes composing it. The box groups were then assigned to one of two sets based on their location, dividing the Bronx into two disjoint regions. The buildings in Region 1 (the South Bronx) are older, the region is more densely populated, and it has a very high rate of fire alarms. Region 2 has a lower population density and fewer fire alarms. In both, about 40 percent of the street box alarms are false while less than 5 percent of the telephone alarms are false. Some other regional characteristics are:

Table 1. INCIDENT CHARACTERISTICS

	No. of units required		Average work times (mins.)		Percentage of all alarms in region (1968)			
					Region 1		Region 2	
Incident type	Engines	Ladders	Engines	Ladders	Box	Phone	Box	Phone
False alarms	1	1	5	5	24.2	2.3	12.6	3.0
Easy emergencies, non-structurals, and transportation fires	1	1	18	18	23.9	15.7	12.0	29.3
Hard emergencies and easy structural fires	1	1	18	18	13.1	17.1	7.4	32.1
Structural fires	2	1	75,45	60	1.02	1.08	.51	1.53
Structural fires	3	2	150,105,60	150,90	.60	.63	.30	.90
Structural fires	7	3	240,180,120,90, 90,60,60	180,135,105	.12	.13	.06	.18
Structural fires	11	4	360,300,270,240, 240,180,150,120, 120,90,90	330,270,180, 135	.06	.06	.03	.09
Total					63%	37%	33%	67%

Region 1

- over 2/3 of the alarms in the Bronx
- covers 1/4 of the borough's area
- almost 2/3 of the alarms in the region are reported by street box

Region 2

- less than 1/3 of the alarms in the Bronx
- covers 3/4 of the borough's area
- 1/3 of the alarms in the region are reported by street box

The exact proportions used in the experiments were based on analysis of 1968 incidents. A percentage breakdown of incidents by type for each region is included in Table 1.

Since modelling the incidence of box and telephone alarms for each type of incident at the location of each box group as independent Poisson processes yields good fits to the observed data (see [3]), we have used this Poisson assumption in generating alarms. For computational ease, we generate independent exponential random variables for the times between successive incidents in the entire borough. The type and location of each incident is then determined by matching random numbers to conditional probabilities. Specifically, for each incident, we let it happen at a particular alarm box with probability equal to the proportion of all 1968 Bronx incidents which occurred there. This location also determines the region in which the incident occurs. Given the region assignment, we let the incident be a box or telephone alarm with the probability appropriate to the region. The incident is then assigned a type using a probability appropriate to the region and how it was reported.

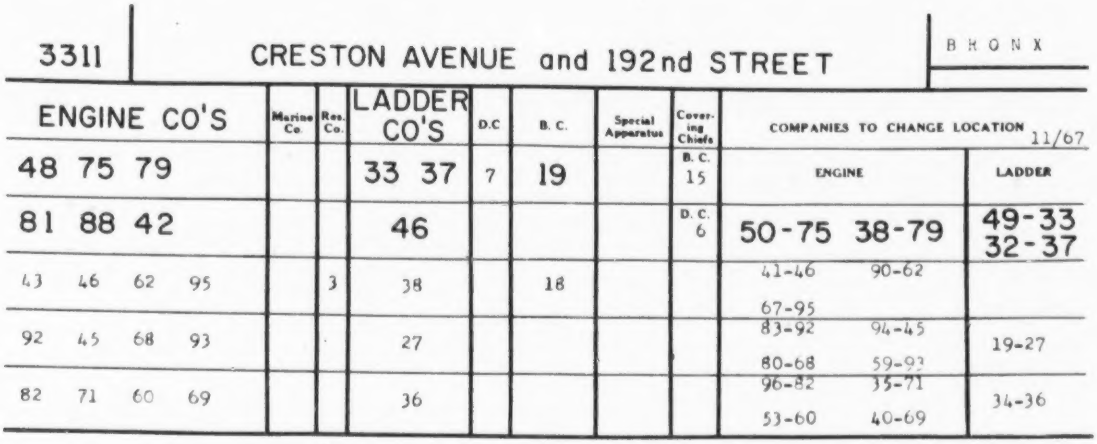
IV. POLICY ANALYSIS

In the last three years we have made many simulation runs to evaluate the effects of various deployment policies. Many of them have led directly to the implementation of new policies by the New York City Fire Department; others have indicated deployment changes which are now being considered. We describe some of these simulations in this section.

A. INITIAL DISPATCH AND ALLOCATION POLICIES

This series of simulation experiments considered a way of reducing the heavy workload being experienced by some companies, without either sacrificing fire-fighting effectiveness or making a large investment in new fire companies. The solution examined consisted of adding a small number of new full-time or part-time companies and modifying the dispatching policy to send fewer companies to some alarms. The potential locations for the new units were existing fire houses. The locations used were determined by practical conditions, such as space in the fire house for men and equipment (part-time units needing less), and the need for help, as measured by the workload of the current units.

New York City's dispatching policies employ alarm assignment cards, one of which is associated with every alarm box in the City (a sample card is shown in Fig. 1). The first half of the card lists the engine companies and ladder companies in increasing order of distance from the



3311 CRESTON AVENUE and 192nd STREET BRONX

ENGINE CO'S	Marine Co.	Res. Co.	LADDER CO'S	D.C	B. C.	Special Apparatus	Covering Chiefs	COMPANIES TO CHANGE LOCATION 11/67	
48 75 79			33 37	7	19		B. C. 15	ENGINE	LADDER
81 88 42			46				D. C. 6	50-75 38-79	49-33 32-37
43 46 62 95		3	38		18			41-46 90-62 67-95	
92 45 68 93			27					83-92 94-45 80-68 59-93	19-27
82 71 60 69			36					96-82 35-71 53-60 40-69	34-36

Fig. 1. A Typical Alarm Assignment Card

alarm box. The traditional dispatching policy for box alarms is to send whoever is available of the first three engines and two ladders listed on the card for the box, "special calling" companies if necessary to assure a response of at least one engine and one ladder. The new dispatching policy to be tested, called adaptive response (AR), would send exactly two engines and one ladder to alarms reported from selected street boxes. (In the simulation, the change to AR required only the rewriting of the ALARM subroutine.)

We simulated because naive calculations of the effects of these proposals were inadequate and simulation would permit more precise calculations. For example:

(1) Under the traditional dispatching policy as few as one engine and one ladder might be sent to a box alarm because other units were unavailable. Therefore, the effect on workload of adding units without changing the response policy was hard to calculate. The same work would not be split among a larger number of units. Instead, more units would be available on the average, so the total number of responses would go up, not necessarily reducing any company's workload.

(2) Sending exactly two engines and one ladder under adaptive response might not reduce any company's workload either. In the case of engines, for example, even though some box alarms received three engines under the traditional policy some also received only one. Thus, it was even possible that adaptive response, at least during busy periods, might actually increase the total number of engine responses.

(3) The effect of adaptive response on response time was also unclear. If availability were increased, then reductions in first and second engine and first ladder response times would be expected. If a third engine or second ladder were needed at an incident at an AR box, one would guess that its response time would go up, since these alarms would always have to wait for the first arriving unit to request additional help. However, the overall average third engine or second ladder response time could end up being reduced since, even under adaptive response, telephone alarms which sound serious are dispatched

> the full complement of three engines and two ladders. Reducing response to potentially less serious box alarms means a greater chance of having a nearby third engine available for a serious telephone alarm. Also, even though the initial dispatch of the third engine (when needed) at AR boxes is delayed by the amount of time it takes the first unit to arrive, the third engine will, on the average, be closer than under the traditional policy and might sometimes get to the fire faster.

We simulated the adaptive response policy for several different specified numbers of fire companies and four different alarm rates, 5, 13-1/3, 21 and 30 alarms per hour in the borough. These alarm rates roughly correspond to the average alarm rates for early morning, midday, evening and a peak evening. All runs at the same alarm rate used the same sequence of incidents (numbering about 2,000), so that true differences between policies were not obscured by their facing different alarm realizations. The results are given in Tables 2 and 3.

The most important interpretation of these results is that, at high alarm rates, the adaptive response policy apparently dominates the traditional one for ladders. That is, we see that at 30 alarms per hour and 12 ladders, the average time to first and second ladder both decrease, and the responses per hour per ladder decrease. (We say apparently because neither response time reduction is, by itself, statistically significant.)

For engines, we see that under adaptive response at either of the two high alarm rates, all three response times decrease (with a statistically significant reduction in second engine time). However, the average number of engine responses per hour increases, which implies that engine availability would be so low under the traditional policy that, on the average, fewer engines were dispatched than under adaptive response.

From Tables 2 and 3 we also note that under either the traditional policy or AR, adding new units has a greater effect on response times at high alarm rates than at low ones. Under the traditional policy, for example, reduction in the average first ladder response is about one second (.025 minutes) per ladder added at 5 alarms per hour; 5 seconds per ladder added at 13.5 alarms per hour; and 16 seconds per ladder added at 30 alarms per hour.

As we supposed, the workload reductions for busy units when companies are added under the traditional response policy turn out to be less than what might naively be expected. For example, for ladders at 30 alarms per hour, we have 2.072 responses per ladder per hour with 12 ladders. When three ladders are added, if the same work were to be redistributed, we would expect (12/15) x 2.072 or 1.658 responses per ladder per hour. However, the simulation results in 1.942 responses per ladder, indicating that the main effect of the new ladders on the original ones is to make them available to answer alarms that previously received one ladder or a ladder from outside the region.

Table 2. ADAPTIVE RESPONSE SIMULATION TEST
REGION 1 RESULTS: ENGINES

Bronx alarm rate (alarms/hr.)	No. of engines	Response times (mins.) to (without AR/AR)			No. of responses/hr. per engine
		First engine	Second engine (when needed)	Third engine (when needed)	
5	18	2.30/	3.26/	4.32/	.533/
	19	2.30/	3.26/	4.28/	.474/
13-1/2	18	2.56/2.55	3.55/3.43	4.81/5.35	1.174/1.079
	19	2.53/2.52	3.53/3.39	4.79/5.25	1.136/1.028
	20	2.44/2.42	3.43/3.37	4.75/5.27	1.102/.986
	21	2.41/2.39	3.42/3.33	4.72/5.16	1.068/.943
21	18	2.92/2.89	4.47/4.07	6.13/6.05	1.649/1.657
	21	/2.62	/3.78	/5.80	/1.468
30	18	3.57/3.57	6.12/5.33	8.07/8.05	1.829/2.224
	21	3.13/3.10	5.05/4.62	6.76/6.75	1.940/2.041
Range of no. of incidents		1820-2208	50-78	25-31	
Range of raw std. dev. of indicated response times		.81-1.93	1.72-2.39	2.56-2.74	

Table 3. ADAPTIVE RESPONSE SIMULATION TEST
REGION 1 RESULTS: LADDERS

Bronx alarm rate (alarms/hr.)	No. of ladders	Response times (mins.) to (without AR/AR)		No. of responses/hr. per ladder
		First ladder	Second ladder (when needed)	
5	12	2.58/	4.02/	.547/
	14	2.53/	3.81/	.480/
13-1/2	12	2.93/2.90	4.42/5.07	1.196/.969
	13	2.79/2.77	4.19/4.90	1.143/.900
	14	2.79/2.78	4.01/4.80	1.086/.846
	15	2.67/2.66	3.90/4.61	1.046/.794
21	12	3.47/3.44	5.67/6.12	1.678/1.473
	15	/2.99	/5.44	/1.238
30	12	4.45/4.37	8.11/7.94	2.072/1.927
	15	3.61/3.55	6.82/6.72	1.942/1.710
Range of no. of incidents		1820-2211	50-78	
Range of raw std. dev. of indicated response times		.97-2.07	1.94-3.41	

Tables 2 and 3 can also be used to compare the benefits derived from adding part-time companies to those derived from adding full-time companies. For example, assuming that a day consists of three 8-hour periods at each of the last three alarm rates (with these high alarm rates meant to correspond to what can be expected in the near future), we can compare adding one 24-hour unit to adding three that work only eight hours each evening. The average daily workload would be 35 responses per ladder and 40 per engine. Under AR, the reduction in average daily responses per region 1 ladder would be about 1.67 if one ladder is added around the clock (1.67 = 8 hours x $\sum_{i=1}^{3}$ (responses/hr/ladder in period i with 12 ladders - responses/hr/ladder in period i with 15 ladders)/(15 - 12) = 8 x (1/3)[(.969 - .794) + (1.473 - 1.238) + (1.927 - 1.710)]). If three ladders were added in the evening, the reduction would be about 1.74 response per ladder per day (= 8 x (1.927 - 1.710)). A similar calculation for engines shows that the reduction in average daily responses for region 1 engines is about 1.35 for one full-time engine and about 1.46 for three evening only engines.

Overall, we see that by adding new companies and using adaptive response during the evening hours we can get both a reduction in company workload and an improvement in average response time relative to the traditional policy. Encouraged by the results of these simulation experiments, the New York City Fire Department adopted AR in part of region 1 in the evenings and added several part-time and full-time fire-fighting units in late 1969.

B. RELOCATION POLICIES

One aspect of deployment is the relocation of available fire companies to fill holes in coverage created when one large fire or several small fires are being fought simultaneously in a single area of the city. Currently in New York City the alarm assignment cards are used to specify predetermined relocations based upon houses made empty when companies are working at an alarm at a particular box. (See Fig. 1. The right-hand side of the card lists the relocations.) The relocations specified are based on the assumption that the alarm at that box is the only alarm in progress in the general area and, therefore, that each company specified to relocate is available to do so.

This method of pre-planned relocations breaks down when, as is an increasingly common occurrence, several incidents are in progress simultaneously in one area. An algorithm was developed [4] which replaces the system of predetermined relocations by a system which determines relocations based on current information on incidents in progress and current unit availability. It was designed for use in the Department's new on-line computerized Management Information and Control System.

We used the simulation to aid in designing the new algorithm and to compare the performance of the new algorithm to the system currently being used. The input program was used to prepare a sequence of 3620 incidents covering a 180-hour period of constant high alarm rate--equivalent to three weeks of evening periods placed end to end. We did not want to look at low alarm periods of the day since few relocations would be required during these periods, and little difference could be seen between policies. Again, for control purposes, the same sequence of incidents was faced by both policies. The adaptive response policy described above was used to determine the initial dispatch to alarms.

We compared the results of these two simulations using three different measures of performance: coverage, workload, and response times.

Coverage

Fire is a random phenomenon, and since the Department cannot be sure where the next alarm will come from, it tries to position companies so that, no matter where the next fire occurs, there will be units available close by. The fire houses located throughout the city provide this protection when they are occupied, but, when fires are in progress, some houses become empty and the "coverage balance" is upset. Relocations are used to correct the imbalance. This is the most important reason for making relocations.

We measured the degree to which the current and the proposed relocation policies succeeded in providing adequate coverage by sampling, at 15-minute intervals, the proportion of alarm boxes which had at least one of their two closest ladder companies available and the proportion of boxes which had at least one of their three closest engine companies available. For coverage purposes the higher the proportions the better the relocation policy.

We found that the proposed algorithm was able to improve coverage considerably. Table 4 presents the empirical cumulative distribution functions for the coverage measure under each of the relocation policies.

Table 4. RELOCATION SIMULATION TEST RESULTS: EMPIRICAL CUMULATIVE DISTRIBUTION FUNCTIONS FOR COVERAGE

$F(x) = P$ (<x percent of boxes have at least 1 of 2 closest ladders available)

x	$F_C(x)$ (Current policy)	$F_P(x)$ (Proposed policy)
35	.000	.000
40	.001	.000
45	.006	.001
50	.007	.006
55	.011	.006
60	.024	.008
65	.043	.018
70	.063	.032
75	.096	.064
80	.152	.113
85	.257	.195
90	.423	.338
95	.622	.555
100	.944	.944

The empirical cdf evaluated at x is the proportion of the samples at which <x percent of the alarm boxes have at least one of their two closest ladders available. The lower the value of this proportion for a given value of x, the

better the policy that produced it is for coverage. Letting $F_C(x)$ be the empirical cdf from the current relocation policy and $F_P(x)$ be the empirical cdf from the proposed relocation policy, we see from Table 3 that

$$F_C(x) \geq F_P(x) \text{ for all } x.$$

These results indicate that, as far as coverage is concerned, the proposed relocation policy is at least as good as the current policy and may be considerably better. The proportion of the time that less than 90 percent of the boxes have at least one of their two closest ladders available drops from .42 to .34, a 19 percent decrease.

Response Time

In the proposed relocation algorithm coverage is the criterion used to determine which of the empty houses should be filled. To determine which of the available companies should be moved to fill those houses a response time criterion is used. Of course, coverage and response time are related measures; coverage representing latent response times. But they are far from equivalent.

The proposed relocation algorithm led to small but consistent improvements in response time. For example, we found a 1 percent reduction in average time to first arriving ladder and a 6 percent reduction in average time to the second ladder. (The simulation runs were too short for these and similar differences to be statistically significant.)

The behavior in the tails of the response time distributions is also important. It was hoped that the new relocation policy would reduce the probability of having large response times to fires. The simulation indicates that this will happen:

Let $G_C(x)$ = fraction of all alarms which have a first ladder response time $\leq$ x minutes using the current relocation policy

$G_P(x)$ = fraction of all alarms which have a first ladder response time $\leq$ x minutes using the proposed relocation policy.

We found that $G_C(x) \leq G_P(x)$ for all values of x.

The inequality also holds when we look at second ladder response times rather than first ladder. Differences between the two policies are greater than for first ladders and are often meaningful. For example, the proportion of alarms at which the second ladder arrives in more than 8 minutes was reduced from .15 to .08.

Workload

There is a wide disparity in workload among fire-fighting companies in New York City. There are areas of the city in which companies respond to over 6000 alarms a year, while several miles away other companies respond to fewer than 2000 alarms. Relocation, as an auxiliary effect, can help balance workload. Generally, the empty houses to be filled will be the houses of the busy companies. If a company's workload were considered in choosing companies to fill these houses, low running companies could be given increased work. The current relocation policy does not consider workload; the proposed one does.

The simulation showed that the proposed policy would lead to significant shifts in workload. For example, one busy company's workload was reduced 10 percent while a slow low running company's workload was increased by 17 percent.

After these initial tests of the proposed relocation policy were made, to determine if the policy would work at least as well as the present policy, we used the simulation again to test the proposed policy on a real scenario which represented the alarms received on one of the worst evenings ever experienced in the Bronx. No change was required in the simulation program to test this case; it was sufficient to bypass the input program and use the real data as input to the simulation program. We then compared the simulation's output with the actual results from that evening. The results are reported in [4].

C. OTHER ALLOCATION POLICIES

One approach to matching the fire-fighting resources on duty during a given period to the demand for their services is to create "part-time" companies which operate only during the periods of high alarm incidence. This approach was described above. An alternative approach is to use existing full-time units, but revise the firemen's work hours to have the on-duty periods for the two shifts (or platoons) of firemen overlap or run concurrently.

Currently one platoon of firemen works from 9 a.m. to 6 p.m. (9 hours) and the second platoon works from 6 p.m. until 9 a.m. (15 hours). Under the proposed work-chart (called the "concurrent two platoon" schedule) one platoon would work from 3 p.m. to midnight (9 hours) and the second platoon (manning a separate piece of apparatus) would work from 9 a.m. to midnight (15 hours). This schedule would place on duty in the concurrent company's house the following number of units over the day:

midnight-9 a.m.	0 units
9 a.m.-3 p.m.	1 unit
3 p.m.-midnight	2 units

Assuming there are nearby units manned in the usual way, this schedule provides a time distribution of units which closely matches the time distribution of alarms. For example, two units in an area, one a concurrent, would provide 1, 2, and 3 units on duty in the area in the three time periods.

The impact of such a reassignment of manpower was hard to predict without simulation, particularly if it were tried in conjunction with a change in the response policy for street box alarms (namely a change to the adaptive response policy previously described). Between 9 a.m. and 3 p.m. there would be no difference in workload or availability since there would be no change in the number of active companies. However, between midnight and 9 a.m. there would be fewer units on duty than before, and the increase in response time to alarms occurring during these hours would have to be compared to the reduction in workload and response times produced during the hours of 3 p.m. to midnight.

In addition, the magnitude of increase in workload for the concurrent company would have to be examined.

We ran the simulation using three different alarm rates:

- o 5 alarms per hour, representing the period midnight to 9 a.m. (period 1)
- o 10 alarms per hour, representing the period 9 a.m. to 3 p.m. (period 2)
- o 20 alarms per hour, representing the period 3 p.m. to midnight (period 3)

The traditional initial dispatch policy was used for rates of 5 and 10 alarms per hour and the adaptive response dispatch policy was used for 20 alarms per hour (since adaptive response had been shown to be most effective at high alarm rates and when extra companies were on duty). The number of ladders on duty in region 1 (the same high activity region as in the first set of simulations) was varied from 10 to 19 and the number of engines from 12 to 25. The simulation was run for 22 different combinations of alarm rate and number of units on duty. In each case a run length of 3620 incidents was used. The same sequence of incidents was used for the 5 and 10 alarms per hour runs (although the alarms were occurring faster at 10 alarms per hour). But, for 20 alarms per hour, a different probability distribution of incident types was used (since false alarms and rubbish fires represent a significantly higher percentage of alarms in the evening than at other times).

Results from 16 of the simulation runs are presented in Table 5. They may be used in several ways to develop allocation policies which match service to demand. For example, suppose we wish to determine the number of concurrent companies to create so as to minimize the average response time to all serious fires (fires requiring the services of two or more ladder companies). The rate of occurrence of serious fires is not the same throughout the day. The percentage of all alarms in region 1 which are serious, the average number of serious alarms which occur per hour, and the percentage of the day's serious alarms which occur in each of the 3 periods of the day are given below:

Period (i)	Serious Percentage	Serious alarms per hour	Percentage of day's serious alarms
1	5	.25	22
2	4	.40	24
3	3	.60	54

To evaluate alternative concurrent two-platoon policies we look only at allocations which leave 21 engines and 15 ladders in region 1 (the number currently located there) between 9 a.m. and 3 p.m., take away a certain number of units (n) between midnight and 9 a.m., and add this number of units to the region for the 3 p.m. to midnight shift. For example, one such policy (for n = 3) locates, respectively, 18, 21 and 24 engines in region 1 for the three periods of the day (which means transforming three full-time engine companies into concurrent two-platoon companies).

To determine the n which minimizes the average first engine response time to serious fires in region 1 we calculate, for each n:

Table 5. CONCURRENT TWO-PLATOON SIMULATION TEST
REGION 1 RESULTS

	Engines					Ladders			
Alarm rate (alarms/hour)	No. in region	Workload (responses/hour)	Response times (mins.) 1st E	Response times (mins.) 2nd E	Response times (mins.) 3rd E	No. in region	Workload (responses/hour)	Response times (mins.) 1st L	Response times (mins.) 2nd L
5 (midnight-9 a.m.)	12	.720	2.41	3.80	4.90	10	.623	2.57	4.17
	14	.642	2.31	3.51	4.74	11	.577	2.47	3.99
	17	.538	2.28	3.51	4.33	12	.549	2.46	3.87
	18	.515	2.20	3.25	4.17	13	.515	2.40	3.72
	21[a]	.448	2.17	3.17	3.83	15[a]	.443	2.33	3.53
10 (9 a.m.-3 p.m.)	12	1.261	2.76	4.47	5.86	10	1.119	2.94	4.92
	14	1.152	2.58	4.10	5.39	11	1.050	2.78	4.67
	17	.996	2.45	3.85	4.80	12	1.018	2.73	4.44
	18	.956	2.33	3.61	4.71	13	.962	2.62	4.29
	21[a]	.849	2.27	3.43	4.16	15[a]	.838	2.47	3.84
20 (3 p.m.-midnight)	12	2.075	3.45	4.79	7.61	10	1.631	3.66	6.51
	14	1.958	3.15	4.45	6.58	11	1.541	3.43	6.16
	21[a]	1.464	2.50	3.62	5.31	15[a]	1.283	2.77	5.04
	24	1.308	2.35	3.32	4.82	17	1.158	2.66	4.75
	25	1.260	2.32	3.29	4.90	18	1.105	2.56	4.64
						19	1.053	2.50	4.50

[a] The number of units currently located in Region 1.

$$S(n) = .22R_1(21-n) + .24R_2(21) + .54R_3(21+n)$$

where $S(n)$ = the average first engine response time to serious fires with n concurrent engine companies

$R_i(k)$ = the average first engine response time in time period i with k engine companies located in region 1.

The values of $S(n)$ for n = 0, 3 and 4 are tabulated below:

n	S(n)
0	2.372 minutes
3	2.298
4	2.299

We see that creating 3 (or 4) concurrent engine companies would reduce first engine response time to serious fires by about five seconds.

The use of concurrent companies has another effect on response times, serving to reduce the wide spread in average response times over the day. For example, in the above case (n = 3 for engines) the spread in average response time to the third engine is reduced from 1.48 minutes (n = 0) to .66 minutes without seriously degrading response times during the early morning hours (the first and second engine response times during period 1 remain better than during either of the other periods).

The other major effectiveness measure which is affected by the creation of concurrent companies is workload. Continuing the example used above, suppose 3 concurrent engine companies are created. As a result, 18 engines remain on duty in period 1 with each one making more responses. The average number of runs made per engine company during this period is increased from 4.0 to 4.6. But, by having 24 units on duty in period 3 instead of 21 the average number of runs per engine company during this period is reduced from 13.2 to 11.8, lightening their burden during the busiest time of day, when they need it most.

Although it is interesting to look at the effects of concurrents on average workload, their principal usefulness is to reduce the workload of specific busy companies. The workload effect from creating concurrents will vary widely depending on which companies are chosen to become concurrents where they are deployed and what the response policy is. If the companies made into concurrents are low running companies, and they are stationed with high running companies, splitting their responses to alarms, then a better balance in company workload would be obtained.

The simulation can be used to assess the workload effect of concurrents on each individual company. In particular, Table 6 shows the distribution of work among the 21 engine companies in Region 1, and the effect of creating three concurrent engine companies. The average number of responses per hour has been tabulated for each company for each time period with and without the three concurrent companies. For each company the average number of daily responses has been calculated from the hourly averages. The three companies chosen to be made into concurrents are each co-located with another engine company at present, so no engine house was left vacant during period 1. Their partners are indicated by a single asterisk in Table 6.

The results indicate that several of the engine companies (principally those which gain a partner during period 3, indicated by a double asterisk in Table 6) would obtain a significant reduction in period 3 workload. For example, the workload of engine 10 in period 3 is reduced 32 percent, from 1.42 responses per hour to .96 responses per hour. Since it experiences only a small increase in workload in period 1, its average daily responses drop 17 percent, from 23 to 19. However, most of the 18 regular companies do not get so much relief.

The three concurrent companies become hard working companies. Since they do not work during period 1 and double up during period 3 their workload, which had averaged 14 percent less than that of the other 18 companies, becomes almost 16 percent higher than the new reduced average workload of the 18 companies. One concurrent company, which had been the lowest running company of all 21 units, becomes thirteenth lowest. The other two concurrent companies have their rankings increased from 10th to 17th lowest and from 12th to 18th lowest (or fourth highest).

The concurrent two-platoon system has not yet been implemented. There is a natural reluctance of the fire-fighters to change their working hours, which are now guaranteed by a provision in the state constitution. But, chances for implementation grow as the city's budget problems and demands for increased productivity grow.

Table 6. EFFECT ON WORKLOAD OF CHANGING 3 PERMANENT ENGINE COMPANIES TO CONCURRENT COMPANIES

Engine identification	Average number of responses/hour (without concurrents/with 3 concurrents)			
	Midnight-9 a.m.	9 a.m.-3 p.m.	3 p.m.-midnight	Daily average
Regular companies				
1**	.54/.55	.96/.96	1.48/1.08	23.93/20.46
2	.61/.61	1.06/1.06	1.73/1.68	27.39/26.95
3*	.43/.77	.87/.87	1.48/1.39	22.41/24.66
4	.35/.36	.70/.70	1.41/1.32	20.04/19.32
5	.35/.36	.70/.70	1.41/1.32	20.04/19.32
6	.53/.54	.99/.99	1.59/1.49	24.96/24.20
7	.34/.35	.69/.69	1.28/1.23	18.72/18.36
8	.34/.35	.69/.69	1.28/1.23	18.72/18.36
9*	.35/.62	.72/.72	1.48/1.33	20.79/21.87
10**	.51/.52	.96/.96	1.42/.96	23.13/19.08
11	.76/.77	1.31/1.31	1.97/1.99	32.42/32.69
12	.83/.82	1.39/1.39	2.03/2.02	34.09/33.99
13	.30/.31	.60/.60	1.12/1.09	16.37/16.15
14	.45/.46	.87/.87	1.56/1.42	23.31/22.14
15**	.66/.66	1.17/1.17	1.77/1.28	28.88/24.47
16	.37/.37	.74/.74	1.27/1.20	19.20/18.57
17	.37/.37	.74/.74	1.27/1.20	19.20/18.57
18*	.27/.50	.54/.54	1.11/1.04	15.66/17.10
Average: Regular companies	.46/.52	.87/.87	1.48/1.35	22.74/22.01
Concurrent companies				
19a	.27/--	.54/.54	1.11/1.04	15.66/24.12
b			--/1.28	
20a	.35/--	.72/.72	1.48/1.33	20.79/26.01
b			--/1.08	
21a	.43/--	.87/.87	1.48/1.39	22.41/26.37
b			--/.96	
Average: Concurrents	.35/--	.71/.71	1.36/2.36	19.62/25.50

*Presently one of the 3 companies to be made into a concurrent company is co-located with this company and shares its responses. When three concurrent companies are created one of them will be co-located with this company for 15 hours (9 a.m.-midnight) and will share its responses.

**When 3 concurrent companies are created one of them will be co-located with this company for 9 hours (3 p.m.-midnight) and will share its responses.

REFERENCES

1. Blum, E. H., "The New York City Fire Project," in Analysis of Public Systems, A. Drake, R. Keeney, P. Morse (eds.), M.I.T. Press, 1972.

2. Carter, G., and E. Ignall, "A Simulation Model of Fire Department Operations," IEEE-System Science and Cybernetics, Vol. 6, No. 4, October 1970.

3. Chaiken, J., and J. Rolph, "Predicting the Demand for Fire Services," The New York City-Rand Institute, P-4625, May 1971.

4. Kolesar, P., and W. Walker, "An Algorithm for the Dynamic Relocation of Fire Companies," The New York City-Rand Institute, R-1023, 1972.

ON-LINE SIMULATION OF URBAN POLICE PATROL AND DISPATCHING

Richard C. Larson

Massachusetts Institute of Technology

Abstract

This paper describes a computer simulation of police patrol forces that has been implemented for resource planning in several police departments. The work is based on the simulation methodology described in Urban Police Patrol Analysis (M.I.T. Press, 1972). Accompanying the presentation will be an on-line computer demonstration of the model using a data base supplied by the Boston Police Department. The developed system is general and can be adapted to suit the needs of any police department in evaluating policies in the following areas:

- the allocation of preventive patrol effort and the effect of changes in patrol resources and manpower scheduling on the allocations.
- the design of standard or overlapping sectors.
- the costs and benefits of an automatic car locator system.
- response patterns for specialized units (e.g., police ambulances).

I. Introduction

Until very recently police departments did not have access to quantitative decision-aiding tools that have gained wide acceptance in industrial and military settings over the past two decades. Prior to the work of the President's Commission on Law Enforcement and Administration of Justice[1], the urgent need for these tools was not widely known. The Commission's recommendations and the 1968 Omnibus Crime Control and Safe

Street Act[2] provided the impetus for research and development to assist police administrators in addressing a wide range of important policy questions:

- Is a ten percent increase in manpower justified?
- What are the tradeoffs between the activities of responding to calls and performing preventive patrol?
- How is an automatic car locator system to be evaluated?
- What would be the effects of shifting to one-man cars in parts of the city?
- Should the tour structure be changed?
- Should dispatching procedures become more formalized?
- How should sectors be designed?
- If ambulance runs were made the responsibility of police, how would overall performance be altered?

That these questions were not receiving systematic attention is evidenced by the fact that far less than one percent of the budgets of police departments had been devoted to research or development and that usually 90 percent or more of the costs of a police department were consumed directly by salaries and fringe benefits.

One response to these needs is the recent development and implementations of a general purpose simulation model of police dispatch and patrol operations. This model is constructed to allow its users to replicate to a very great extent the actual dispatch and patrol operations of most urban police departments, thereby providing a tool to assist in answering the types of questions listed above. Police administrators should find simulation models valuable for the following purposes:

1. They facilitate detailed investigations of operations throughout the city (or part of the city);
2. They provide a consistent framework for estimating the value of new technologies;
3. They serve as training tools to increase awareness of the system interactions and consequences resulting from every day policy decisions;
4. They suggest new criteria for monitoring and evaluating actual operating systems.

A recent article by Colton[3] reporting survey results from approximately 500 police departments revealed that police themselves view the use of computers for resource allocation as the single most important application of computers in the coming years. Simulation models and other analytical tools should play an important role in this work.

This paper will outline the structure of the model developed by the author, its use in an online interactive mode, and its current implementation status in several large U.S. cities. Accompanying the oral presentation of the paper will be a demonstration of the model, using data derived from the implementation at the Boston Police Department (Boston, Massachusetts).

II. Overall Model Structure

The simulation works in the following way: Incidents are generated throughout the city, distributed randomly in time and space according to observed statistical patterns. Each incident has an associated priority number, the lower numbers designating the most important incidents. For instance, a "priority 1" incident would be "officer-in-trouble," "felony-in-progress," or "seriously injured person;" a "priority 4" incident could be "open fire hydrant," "lock-out," or "parking violation." As each incident becomes known, an attempt is made to assign (dispatch) a patrol unit to the scene of the incident. In attempting this assignment, the computer is programmed to duplicate as closely as possible the decision-making logic of an actual police dispatcher. In certain cases this assignment cannot be performed because the congestion level of the force is too high; then, the incident report (which might in actuality be a complaint ticket) joins a queue of waiting reports. The queue is depleted as patrol units become available.

The model is designed to study two general classes of administrative policies:

1. The patrol deployment strategy
2. The dispatch and reassignment policy.

The patrol deployment strategy determines the total number of patrol units, whether units are assigned to non-overlapping sectors, which sectors constitute a geographical command, and which areas are more heavily patrolled than others. The dispatch and reassignment policy specifies the set of decision rules the dispatcher follows when attempting to assign a patrol unit to a reported incident. Included in the dispatch policy are the priority structure, rules about cross-precinct dispatching, the queue discipline, and so forth.

There are several important measures of operational effectiveness that the model tabulates. These include statistics on dispatcher queue length, patrol travel times, amount of preventive patrol, workloads of individual patrol units, the amount of intersector dispatches, and so on.

The simulation program is organized to reflect the spatial relationships inherent in patrol operations, as well as the sequential time nature of events which is common to all simulations. First the spatial or geographical structure is discussed, then the time sequence of events.

II. 1. Geographical Structure

The city, or arbitrary shape, is partitioned into a set of "geographical atoms." Each atom is a polygon of arbitrary shape and size. The atoms are sufficiently small so that any probability density functions over the atom (depicting, for instance, the positions of reported incidents) can be considered uniform over the atom. This does not restrict accuracy of results, because the atoms can be arbitrarily small.

A patrol unit's sector is a collection of atoms. The atoms in the collection need not be contiguous (spatially) or consecutive (in the numerical ordering of atoms.) In general, each atom may belong to any number of (overlapping) patrol sectors.

A patrol command (for instance, "precinct," "district," or "division") is also a collection of atoms. Each sector must be fully contained within a command.

The technique that is essential if one is to structure the geographical data in this way is the point-polygon method. This method provides a computer algorithm for answering the following question: "Given a point (x,y) and a polygon specified by its I clockwise ordered vertices $(x_I, y_I), (x_2, y_2), \ldots, (x_I, y_I)$, is the point (x,y) contained within the polygon?" The basic idea of the method, which is fully discussed by S. Nordbeck[4], is to extend a ray in any direction from the point in question; if the ray intersects the sides of the polygon an odd (even) number of times, the point is (is not) within the polygon. The method is completely general and does not require any special properties (for example, convexity) of the polygon. It is particularly well suited for machine implementation, since the tests for intersection are quickly performed on a computer.

In the simulation model the point-polygon method provides a convenient way to generate samples (x,y) uniformly distributed over a geographical atom. The atom, which is a polygon of arbitrary shape, is enclosed in the smallest rectangle fully containing it. Then, using two random numbers, a candidate point that has a uniform distribution over the rectangle is obtained. If this point is also within the polygon, it is accepted as the sample value; otherwise it is rejected and new points generated until one is accepted. The probability that any candidate point will be accepted is equal to the ratio of the area of the polygon (A_P) to the area of the rectangle (A_R). The number of candidate points that have to be generated until one is accepted is a geometrically distributed random variable with mean A_R/A_P. For reasonably compact polygons, this number, reflecting sampling efficiency, is usually less than 2 (and often quite close to 1).

II. 2. Time Sequence of Events

The simulation is an event-paced model. That is, once a certain set of operations associated with one event is completed, the program determines the next event that occurs and updates a simulation clock by adding to the present time the time until the next event. The program then proceeds with the set of operations associated with that event. Once the clock reaches some maximum time (T_{max}), the simulation is terminated and summary statistics are tabulated and printed out. One completed run of the simulation entails inputting data, initialization of simulation status variables, executing the program for an equivalent time T_{max}, and printing the summary statistics.

We do not have space here to provide details of the various dispatching algorithms or patrol deployment policies, but we provide a brief discussion of the important parameters at each point in the simulation.

The main type of event that occurs is a reported incident or a "call for police service." The times of occurrence of calls are generated as in a Poisson process with rate parameter LAMBDA (=average number of calls per hour). The greater the value of LAMBDA, the more likely it is that the system will incur congestion (saturation) of resources. The location of the call is determined from historical patterns which indicate the fraction of calls that originate from each atom; given the atom of the call, its spatial location within the atom is assumed to be uniformly distributed. The priority of the call is determined from historical data which may vary by atom.

Once the position and priority of the incident are known, the program executes a DISPATCH algorithm that attempts to assign a patrol unit to the incident. This algorithm is governed by the dispatch policy specified by the user. One component of the dispatch policy specifies the geographical area from which a unit may be dispatched:

Option 1: Only assign a unit whose patrol sector includes the geographical atom containing the incident (a sector policy)

Option 2: Only assign a unit whose precinct or district designation is the same as that of the incident (a precinct or district policy)

Option 3: Only assign a unit whose division* designation is the same as that of the incident (a division policy)

The particular option on a given run is usually specified at the start of the run, although the user may choose to use the interactive feature to alter the dispatch policy during the course of a run.

Given that a patrol unit is within the correct geographical area for a particular incident, the algorithm then determines whether the unit is considered "eligible for dispatch" to this incident. This determination focuses on estimated travel time to the incident, the priority of the incident, and the current activity of the patrol unit. In general, the user may specify a dispatch policy that allows very important incidents to preempt (interrupt) patrol units servicing incidents of lesser importance. In addition, the "importance" of preventive patrol may vary with each unit, thereby giving the user the capability of assuring at least some minimal level of continuous preventive patrol.

If no unit is found eligible for dispatch, the reported incident is inserted at the end of a queue of other unserviced incidents. There may be separate queues for each command and each priority level.

---

*A division contains several precincts or districts.

If at least one unit satisfies the eligibility conditions, one is selected for dispatch according to a prespecified criterion such as minimal expected travel time. The assigned unit's priority status and position are changed accordingly.

A second major type of event occurs when a patrol unit completes servicing an incident. A REASSIGNMENT algorithm is then executed that either (1) reassigns the returning unit to an unserviced incident or (2) returns the unit to preventive patrol. The eligibility conditions regarding priorities, travel distances, and geographical areas, which are necessary to specify a dispatch policy, are also an integral part of the reassignment policy. In addition, it is necessary to specify how one unserviced incident is given preference over another. This part of the reassignment policy, called the _reassignment preference policy_, parallels the _queue discipline_ in ordinary queuing systems.

II. 3. Location Estimation

If not all available position information is used or if the unit is performing preventive patrol, the method of estimation of patrol unit position must be specified. Three options are available, one which simulates the information provided by an _automatic car locater_ system, and two which simulate estimation guessing procedures that are commonly found today in most police operations.

II. 4. Simulation Variables

The simulation program can tabulate statistics on any algebraically defined variable. The variables that have been most often recorded in our studies are:

1. Total time required to service an incident, that is travel time plus time at the scene.
2. Workload of each patrol unit (measured in total job assignments and in time spent on jobs).
3. Fraction of services preempted.
4. Amount of preventive patrol.
5. Travel time of a unit to reach the scene of the incident.
6. Dispatcher queue length.
7. Dispatcher queue wait.
8. The number of intersector dispatches.
9. The fraction of dispatch and/or reassignment decisions for which the car position was _estimated_, rather than known exactly.
10. The fraction of dispatch decisions which were nonoptimal, in the sense that there was at least one available unit closer to the scene of the incident.
11. The extra distance traveled as the result of a nonoptimal dispatch assignment.

As will be discussed below, each variable may be tabulated at any one of several levels of aggregation.

III. On-Line Interactive Capabilities

During the past two years a great deal of effort by J. Williamson, R. Couper, and

C. Vogel* has been devoted to implementing an easy-to-use on-line Input/Output package with the simulation. This effort has resulted in a program that is readily usable by someone without detailed knowledge of computer operation, the simulation logic, or statistics.

The core of the I/O package is a sequential tree structure that presents to the user the options that are available to him. If the user expresses interest in a particular option, details of use are printed out, the level of which is determined by the responses of the user. Default options are standard, so that if the user does not know what to do at a particular point, a simple carriage return yields additional helpful information. A sample "I/O session" is depicted in Figure 1.

Once the initial I/O session is completed, the user has specified the following: the particular geographical data base he wishes to employ (these data are usually stored on disk), the dispatch procedures, the method of car location estimation, the length of the run, and whether he desires to trace the simulation (and possibly interact with it) while in progress.

Following completion of the simulation, a "LEVEL 1" output is printed. A sample is shown in Figure 2. This contains a small number of highly aggregated statistics describing the run: average travel time, average total response time (including queuing delay), average workloads, etc. The LEVEL 1 output contains no statistical jargon (for instance, "variance" or "sample size") and no program variables. It is self-contained and self-explanatory. We have found LEVEL 1 to be quite useful for introducing police planners and administrators to the capabilities of the simulation and for quickly eliminating runs with obviously poor performance characteristics.

At this point the user may request LEVEL 2 output. A sample is shown in Figure 3. As can be seen, this level is less aggregated and provides average values of many variables by priority level. We expect that a sizable number of users will find the information presented in LEVEL 2 adequate for certain high-level planning and decision-making problems (e.g., determining overall manning levels).

If the user desires even more detail, he now requests portions of a LEVEL 3 output. A sample is shown in Figure 4. As one can see, this level presents many detailed statistics and can be of great assistance in very fine-grain planning problems, for instance, sector design. We expect that very experienced users will usually demand LEVEL 3 output before making decisions affecting actual operating procedures in the field or at the dispatcher's position.

Regarding the other on-line capabilities, we have found that the TRACE option (which prints out the details of each call, assignment, and reassignment in real-time) assists new users in learning of the operation of the model and in developing a good intuition for system operation. We also have in mind the use of the TRACE option for training dispatchers in new dispatching procedures. In this mode of operation, the computer would request the user to make the dispatch or reassignment decision at the appropriate times (and the standard DISPATCH and REASSIGNMENT algorithms would be by-passed). Once the "dispatch-user" settles on a particular strategy that he wishes to test in detail, he can stop the TRACE, input the control parameters describing his strategy, and run the model for a sufficiently long time to obtain reliable statistics.

---

*All of Urban Sciences, Inc. of Wellesley, Massachusetts.

IV. Implementations

IV. 1. Boston, Massachusetts

To date, the model has been implemented in detail for the city of Boston[5] and used in a preliminary way in a number of other cities. The Boston implementation requires call-for-service data for each of over 800 "reporting areas" (geographical atoms) and for each of four priority levels. Boston is partitioned into 12 districts (patrol commands), with a total of approximately 90 radio-dispatchable patrol units in the field at any one time. The model has already been used to analyze the effects of various automatic car locator systems for the city. It is currently being used to perform sector redesigns and to determine the effects of adding additional "district-wide" cars to certain districts during heavy workload hours. Deputy Superintendent John Bonner hopes to educate field commanders in its use so that many decisions that are made at the district level could be made with the assistance of the simulation model.

IV. 2. Washington, D.C.

A somewhat different off-line version of the model is being created and implemented for the Washington, D.C. Metropolitan Police Department, under the technical guidance of Mathematica, Inc. and with the support of the Law Enforcement Assistance Administration. Here the city's geographical structure is modeled as a set of discrete points, rather than polygons, each point corresponding to one city (surveyor) block. For Washington, D.C. this represents approximately 6,000 points, or sufficiently fine-grain detail to make the model useful for sector redesigns for the 138 Scout cars distributed throughout the city. The selection of a point geography was based on detailed block-level statistics that are available for Washington, D.C. and on the fact that an off-line model need not produce rapid turn around times (in the same sense as an on-line real-time model). This effort started in January of 1972 and is reported in periodical publications of Mathematica, Inc. and the Washington, D. C. Metropolitan Police Department.

IV. 3. New York City

In August 1972 the New York City Police Department contracted with the New York City Rand Institute to adapt the on-line simulation and related allocation tools* to the special requirements of New York City and to implement these tools for analysis of the entire patrol force (distributed throughout 75 precincts in over 700 regular radio-dispatchable patrol cars, plus special-assignment cars and radio-dispatchable foot patrolmen). The Department hopes eventually to provide each precinct commander with a readily understandable set of on-line decision tools, with easy terminal access from each of the 75 precinct station houses. Thus, as in Boston, it is hoped that these tools will be used for short-term decentralized decision-making, as well as for longer-term, centralized resource allocation and planning and research. As of this writing this work is still in the planning stages, but its progress will be documented in reports from the New York City Rand Institute.

IV. 4. National Research Council of Canada

During the past year or so T. Arnold and F. R. Lipsett of the Radio and Electrical Engineering Division of the National Research Council of Canada have reprogrammed the version of the model detailed in Ref. [7], in order to adapt the programs to their computing system. Their work is currently in progress, aimed at determining the potential usefulness of simulations to small police forces. Recently they have started simulating a co-operating police force near Ottawa which operates with 5 sectors and 5 patrol cars. They anticipate preliminary documentation of this work by January 1973.

IV. 5. Demonstrations in Other U.S. Cities

The New York City Rand Institute, as part of a contract with the U.S. Department of Housing and Urban Development, is demonstrating the use of the on-line simulation model in a number of cities. This is done by identifying cities with expressed interest in quantitative tools to assist planners and decision makers, selecting a subset of these cities, and traveling to the cities with a portable computer terminal which can be connected to the central computer in either Waltham, Massachusetts or San Francisco, California via a simple telephone call into a nation-wide WATS* line network. The long range goal in this work is to assess the usefulness of the model in cities with diverse characteristics, to introduce system planners and decision-makers to the notion of using a simulation model, and to arrive at recommendations for improvement of the model. This work is still in progress and is reported in periodical technical reports published by the New York City Rand Institute.

---

*See, for instance, the resource allocation algorithm described in Chapter 5 of Ref. [8].

*Wide Area Telecommunications Service.

```
ENTER DISTRICTS TO BE SIMULATED (OR ENTER "ALL")
15
ENTER DISTRICTS YOU WISH TO MODIFY
NONE
DO YOU WANT TO CHANGE ANY VARIABLES?
YES
          SIMULATION VARIABLES AND THEIR VALUES
     1.  LENGTH OF SIMULATION RUN = 2.00 HOURS
     2.  NUMBER OF CALLS PER HOUR =
         DISTR.  1   2  3   4  5  6  7  11  13  14  15
         NO.     8  17  8  12  5  6  4  10   5   5   3
     3.  VEHICLE SELECTION METHOD = STRICT CENTER OF MASS
     4.  SERVICE TIME AT SCENE AND VEHICLE RESPONSE SPEED
         PRIORITY               1   2   3   4
         SERV. TIME (IN MIN.)  33  33  33  33
         RESP. SPEED (IN MPH)  15  12  12  10
     5.  TYPE OF SIMULATION OUTPUT = CITY
     6.  MORE DETAILED INFORMATION
ENTER NUMBER(S) OF THOSE TO BE CHANGED

1,3,5
     1.  ENTER THE LENGTH OF THE SIMULATION IN HOURS =
20.
     3.  THERE ARE 3 VEHICLE SELECTION PROCEDURES, THEY ARE =
     1.  MODIFIED CENTER OF MASS
     2.  STRICT CENTER OF MASS
     3.  THE RESOLUTION OF A VEHICLE LOCATION SYSTEM
     PLEASE ENTER THE NUMBER OF YOUR CHOICE =

2
     5.  DO YOU WANT CITY-WIDE OR DISTRICT SIMULATION OUTPUT?
DISTRICT
```

FIGURE 1

SAMPLE I/O SESSION WITH

POLICE DISPATCH AND PATROL SIMULATION

```
STATISTICAL SUMMARIES - DISTRICT NO. 15

THE AVERAGE PATROL UNIT SPENT 34.21% OF ITS TIME SERVICING CALLS

AVERAGE RESPONSE TIME TO HIGH PRIORITY CALLS WAS 6.40 MINUTES

AVERAGE RESPONSE TIME TO LOW PRIORITY CALLS WAS 7.27 MINUTES

AVERAGE TRAVEL TIME WAS 3.19 MINUTES

AVERAGE TOTAL JOB TIME WAS 34.59 MINUTES
```

FIGURE 2

<u>SAMPLE LEVEL 1 OUTPUT OF</u>
<u>POLICE DISPATCH AND PATROL SIMULATION</u>

```
DO YOU WANT TO SEE LEVEL 2 STATISTICS?
YES
STATISTICAL SUMMARIES - DISTRICT NO. 15

AN AVERAGE OF 34.21% OF THE TIME OF ALL UNITS WAS SPENT SERVING CALLS
     THE FOLLOWING UNITS WERE SUBSTANTIALLY BELOW THIS FIGURE:
         UNIT NO.       UNIT TYPE         %
           4             WAGON          0.00

     THE FOLLOWING UNITS WERE SUBSTANTIALLY BELOW THIS FIGURE:
         UNIT NO.       UNIT TYPE         %
           1            SECTOR CAR      79.14

AVERAGE TIMES FOR EACH TYPE OF CALL WERE AS FOLLOWS (STATED IN MIN.)
PRIORITY     DISPATCH DELAY     TRAV. TIME     RESPONSE TIME
    1             0.00              1.60             1.60
    2             5.06              3.40             8.46
    3             0.00              0.00             0.00
    4             3.72              3.55             7.27
ALL CALLS         3.62              3.19             6.81

THE AVERAGE TRAVEL TIME WAS 3.19 MINUTES WITH REGULAR SPREAD
 10.53% OF THE CALLS INCURRED A QUEUING DELAY DUE TO CAR UNAVAILABILITY
  0.32= AVER. EXTRA MILES TRAV. DUE TO DISPATCHING OTHER THEN CLOSEST CAR

THE AVERAGE TOTAL JOB TIME (TRAV. TIME+TIME AT SCENE) BY PRIORITY WAS:
                    1   77.54 MINUTES
                    2   37.45 MINUTES
                    3    0.00 MINUTES
                    4.  18.05 MINUTES

THE AVERAGE QUEUE LENGTH FOR EACH TYPE OF CALL WAS:
                    1    0.00
                    2    0.00
                    3    0.00
                    4    0.00

THE MAXIMUM DELAY IN QUEUE FOR EACH TYPE OF CALL WAS:
                    1    0.00 MINUTES
                    2   35.39 MINUTES
                    3    0.00 MINUTES
                    4   33.46 MINUTES
```

FIGURE 3

SAMPLE LEVEL 2 OUTPUT OF
POLICE DISPATCH AND PATROL SIMULATION
11 C

```
DO YOU WANT TO SEE LEVEL 3 STATISTICS?
YES

DISTRICT SUMMARY

                                        OVERALL  STANDARD  MAXIMUM
       PARAMETER                        AVERAGE  DEVIATION  VALUE
1.  WORKLOAD (%)                         34.2      28.6      79.1
2.  reSPONSE TIME (MINUTES)               6.8      10.9      39.8
3.  TRAVEL TIME (MINUTES)                 3.2       2.0      10.5
4.  EXTRA DISTANCE (MILES)                0.3       0.4       1.2
5.  TOTAL JOB TIME (MINUTES)             34.6      49.2     227.3
6.  NUMBER OF CALLS PREEMPTED FOR HIGHER PRIORITY          =  0 ( 0%)
7.  NUMBER OF CALLS ASSIGNED TO UNIT ON PREVENTIVE PATROL = 17 (89%)
8.  NUMBER OF CALLS ASSIGNED TO UNIT ASSIGNED TO SECTOR   = 17 (89%)
9.  NUMBER OF CALLS ASSIGNED TO CARS OTHER THAN CLOSEST   =  7 (37%)

FOR WHICH PARAMETER DO YOU WANT A FURTHER BREAKDOWN?

                 -----WORKLOAD BY PRIORITY-----
    PATROL UNIT     1       2       3       4       TOTAL
         1        47.4%   17.6%   0.0%   14.2%      79.1%
         2         0.4%   17.3%   0.0%    7.1%      24.8%
         3         0.7%   19.7%   0.0%   12.5%      32.9%
         4         0.0%    0.0%   0.0%    0.0%       0.0%

DO YOU WANT MORE DETAIL FOR ANY OTHER PARAMETERS?
YES
FOR WHICH PARAMETER DO YOU WANT FURTHER BREAKDOWN?
7
BY PRIORITY?
NO
FOR WHICH UNITS?
ALL

CALLS ASSIGNED TO UNIT ON PREVENTIVE PATROL

PATROL UNIT      NO. CALLS      PER CENT
     1               6           100.0%
     2               6            85.7%
     3               5            83.3%
     4               0             0.0%
```

FIGURE 4

SAMPLE LEVEL 3 OUTPUT OF

POLICE DISPATCH AND PATROL SIMULATION

Acknowledgments

Early development of the model reported in this paper was supported in part by the National Science Foundation, through grants to the M.I.T. Operations Research Center; the U.S. Department of Housing and Urban Development, through contracts to the New York City Rand Institute; and the Ford Foundation, through a two-year postdoctoral fellowship. The more recent on-line implementation has been carried out by Urban Sciences, Inc., the work under the supervision of J. Williamson and R. Couper.

Other technical details of the model are found in Reference 8.

REFERENCES

(1) President's Commission on Law Enforcement and Administration of Justice, The Challenge of Crime in A Free Society, Washington, D.C.: U.S.

(2) Omnibus Crime Control and Safe Streets Act, U.S. Public Law 90-351, June 19, 1968.

(3) Kent Colton, "Survey of Police Use of Computers," 1972 Municipal Yearbook.

(4) S. Nordbeck, "Location of Areal Data for Computer Processing," Lund Studies in Geography, Ser. C. General and Mathematical Geography, No. 2, The Royal University of Lund, Sweden, Department of Geography, Lund, Sweden, C.W.K. Gleerup Publishers, 1962.

(5) R. Couper, K. Vogel and J. Williamson, "Final Report on the Computer Simulation of the Boston Police Patrol Forces," Urban Sciences, Inc., Wellesley, Mass., October, 1972.

(6) Insp. H. F. Miller, Jr. and B. A. Knoppers, "A Computer Simulation of Police Dispatching and Patrol Functions," Paper presented at 1972 International Symposium on Criminal Justice, Information and Statistics Systems, New Orleans, Louisiana, October 3-5, 1972.

(7) R. C. Larson, "Models for the Allocation of Urban Police Patrol Forces," Technical Report No. 44, M.I.T. Operations Research Center, Cambridge, Mass., 1969.

(8) R. C. Larson, Urban Police Patrol Analysis, Cambridge, Mass.: M.I.T. Press, 1972.

Session 8: Aerospace Applications
Chairman: Lawrence Heinle, Lockheed Palo Alto Research Laboratory

Quotes from the abstracts. Aircraft track generation and transmission of data to remote sites on real time is discussed in the first paper...The use of this management tool in planning request for proposals is developed in the second paper...Manufacturing use of simulation in Tokyo is illustrated in a new simulator that simultaneously simulates the material flows in the process and the control systems' behavior in the third paper...A probabilistic event store computer simulation of interactions between missile and aircraft is used as text material in the third paper...The Knoxville, Tennessee terminal is used as a testbed for the analysis and design of automated safety separations functions for air traffic control in the last paper in the session.

Papers

"Digital Simulation of a Multiple Element Threat Environment"
K. E. Dominiak, University of Florida
R. J. Ireland, Honeywell Information Systems

"Cost/Resource Model"
Betty J. Lanstra, Hughes Aircraft Company

"MAFLOS--A Generalized Manufacturing System Simulation"
K. Mitome, S. Tsuhara, S. Seki and K. Isoda, Hitachi, Ltd.

"A Description of an AAW Model and Its Classroom Uses"
Alvin F. Andrus, Naval Postgraduate School

Simulation in the Design of Automated Air Traffic Control Functions"
Paul D. Flanagan, Judith B. Currier and Kenneth E. Willis,
Metis Corporation

Discussant: John E. Sherman, Lockheed Missles & Space Company

SIMULATION OF A MULTIPLE ELEMENT TEST ENVIRONMENT

Kenneth E. Dominiak
University of Florida
Eglin Air Force Base, Florida 32542

Ronald J. Ireland*
Honeywell Information Systems
Phoenix, Arizona 85029

Abstract

A real-time simulation of a multiple element defensive test environment is under study and development. This simulation is to be used for evaluating performance of command and control systems. The initial version of the simulation generates simultaneously up to twenty aircraft each with up to two jamming devices. The overall approach is to generate aircraft tracks and jamming returns at a large scale central digital computer. This data is then transmitted to remote radar sites and converted by on-site equipment to analog signals which are injected into the radar circuitry. A unique feature of the approach is that both real and simulated aircraft will be observed and tracked simultaneously by radar operators.

Introduction

A real-time simulation of a multiple element defensive test environment is presently under development. The rationale, design concept, and various features of the simulation are described in this paper with emphasis on the digital portion of the simulation.

Up to twenty aircraft, each with up to two electronic counter-measures (ECM) radar jamming devices, are generated in this initial version of the simulation. Additional numbers of aircraft will be introduced as required, after the basic simulation has been further developed. The overall approach is to generate aircraft tracks and jamming returns at a large scale central digital computer (CDC 6600). This data is then transmitted to remote radar sites and converted by on-site equipment to analog signals which are

---

This work was supported by the Range Development Division of the Air Force Armament Development and Test Center under Contract F08635-72-C-0060.

*Formerly with the University of Florida, Eglin Air Force Base, Florida.

injected into the radar circuitry. A unique feature of the approach is that both real and simulated aircraft will be observed and tracked simultaneously by radar operators. The simulated aircraft will be indistinguishable from real aircraft on the radar displays. Radar operators will then interface with the remainder of the command and control system in a normal fashion.

A simplified block diagram is illustrated in Figure 1. Referring to Figure 1, the topics discussed in this paper include required inputs to the CPU, simulation activities of the CPU such as aircraft track and jamming return generation, transmission of data to the on-site buffers, and feedback information from the radars to the CPU. Not discussed here is the on-site D/A conversion equipment, the radar signal insertion equipment, and the interface between the radars and the command and control center. Both the D/A conversion and signal insertion equipments and associated concepts were developed by the Westinghouse Electric Corp., Baltimore, Md., under a separate contract.

The need for a simulation such as this arises as a result of several factors. A fundamental problem is that of validating a command and control system (CCS) by attempting to defeat it. Each such attempt requires the dedication of large numbers of aircraft, since experience has shown that the relationship between numbers of threat aircraft and utilized capacity of the CCS is highly nonlinear. That is, it is not possible to extrapolate results obtained with a few aircraft to the case of many aircraft (or saturation). As a result, it becomes highly desirable from both economic and operational considerations to generate some of the threat aircraft via simulation. On the other hand, the presence of real aircraft provides increased realism and improves the validity of the exercise. Another somewhat unrelated, but significant, advantage resulting from the use of some real aircraft is that other existing and planned simulations which utilize simulated aircraft only can be validated.

The current stage of development of the simulation can be summarized as follows:

a) Simulation requirements have been established. These are based on numbers of aircraft, maximum aircraft velocities and ranges, types of radars considered, and maximum desirable data rates between CPU and radar sites.

b) Much of the modeling is completed. In some areas, decisions concerning approaches to be taken have not yet been made, however, tradeoff studies are currently underway.

c) On-site digital processing, D/A conversion, signal generation, and signal insertion equipment has been developed and demonstrated.

d) Interface requirements between the CPU and the on-site processing equipment have been established.

e) Input data requirements are only partially defined, and are currently under study.

f) Study of requirements for data

organization, storage, and retrieval is just beginning.

g) Specification of user inputs including format is not yet completed, however, it has been established that a Scenario Input Language will be developed to simplify test specification by users and to maximize control during an exercise.

h) Development of algorithms and coding is just underway. Initial coding will be in FORTRAN IV, however, assembly language will be employed if determined necessary.

Aircraft and Jammer Return Generation

Aircraft track information is provided to each radar in the form of radar cross-section (RCS) and aircraft position in range, azimuth, and elevation. This information must be supplied to the radars by the CPU at a rate sufficient to insure realistic target aircraft behavior on the radar screens, that is, behavior indistinguishable from that of a real aircraft. Three types of radars are initially being utilized - early warning (EW), track-while-scan (TWS), and conical scan (CS). The maximum number of aircraft tracks to be generated for each type of radar are listed in Table I. Aircraft dynamics for the worst case data transfer (i.e., highest data rate) can be bounded using the known maximum target velocity of Mach 3 (3000 ft./sec.) and maximum acceleration of 10g's (322 ft./sec.^{2}). These latter bounds are determined from a consideration of user requirements.

In establishing the basic simulation requirements, it was considered desirable to use existing telephone lines for transfer of data between the CPU and the radar sites. However, the maximum capacity of these lines is 2400 bits/sec. When a maximum data rate was calculated based on the numbers of targets required, the worst case target dynamics, and characteristics of the various radars, it was found to be far in excess of 2400 bits/sec. As a result, it was decided that some computation would be carried out on-site. By reducing the update rate on track parameters from the CPU and performing simple linear interpolation on-site, it was possible to reduce the maximum data rate to 2060 bits/sec.

TABLE I. RADAR TARGET REQUIREMENTS

RADAR TYPE	MAX. NO. OF TARGETS	MAX. TARGET RANGE (MI.)	MAX. JAMMERS PER AIRCRAFT
EW	20	150	2
TWS	8	60	2
CS	4	60	2

All on-site computation is performed using hardware rather than programmable software, and requires no operator to be present.

To illustrate the approach finally adopted, we consider the TWS radar. At intervals of 626 milliseconds, values of target aircraft RCS, range, azimuth, and elevation are made available at the radar site. The number of bits required for each track parameter of each target is shown in Table II. These are determined from accuracy considerations. The 96 bits for "other data" includes jamming information, timing, bookkeeping, and the like. If 8 target tracks are updated at the indicated intervals and with the indicated number of bits, a data rate of approximately 2060 bits/sec. results. However, if the TWS radar display is to appear realistic, the indicated update rate is too slow. Therefore, under the assumption of linear variation of track parameters within a 626 msec. interval, a constant increment is added to each track parameter every 62.6 msec. Thus, the update rate is increased by a factor of 10. The number of bits required for incrementing track parameters is also listed in Table II. Jammer information is provided to the radars in the form of gain factors. These are updated in the same fashion as track parameters.

TABLE II. CPU TO TWS RADAR DATA FLOW

TRACK PARAMETER	NO. OF BITS
RCS	8
RANGE (R)	14
AZIMUTH (θ)	12
ELEVATION (ϕ)	11
ΔRCS	5
ΔR	5
Δθ	5
Δϕ	5
OTHER DATA	96
TOTAL/UPDATE/TARGET	161

Various methods of generating aircraft tracks at the CPU have been examined. At maximum target ranges, it appears that point-to-point trajectories specified in terms of latitude, longitude, and altitude will provide sufficient accuracy and realism. However, at lesser ranges it appears that numerical integration of equations of motion will be required. Several integration methods which are particularly suited to the present application are being examined. Results obtained to date are summarized in the Appendix.

A variety of models and computational schemes either have been developed or are under development for specific target and jammer parameters. However, results are detailed and will not be presented here. Examples of these are the equations used to compute radar returns, models for jammer gains, and algorithms for determining which targets are in the field-of-view of which radars.

Inputs To The Central Processor

As discussed in the Introduction, input data requirements are only partially defined. Problems of data organization, storage, and retrieval will be addressed in the next phase of this continuing effort, as input data requirements become better known. In general, data inputs and file updates will be off-line since requirements for each exercise will be fixed.

Radar antenna patterns are stored at the radar sites rather than at the CPU. As a result, radar returns are computed at the CPU with the target always at mid-beam of the main lobe of the antenna pattern. The return is then modulated by the on-site processing equipment so as to properly position the target within the antenna pattern.

Three-dimensional jammer patterns are stored at the CPU. A requirement for adjacent level differences of .5 to 1.0 decibels in RF signal level has been established for jammer pattern storage. This requirement is based on the conclusion that level differences of this magnitude are essentially indiscernible when observing the resulting video on a radar display.

Each time the simulation is exercised, extensive user inputs in the form of a completely defined scenario will be required. It is important that the specification of a scenario be in terms of parameters and dimensions familiar to the user. It is impractical to expect the user to learn a computer language before he can exercise the simulation. For this reason, it has been determined that a Scenario Input Language will be developed. The language will make it possible for the user to specify aircraft positions, aircraft maneuvers, and the like, in terms familiar to him, thereby rendering the simulation more usable.

Interface With On-Site Equipment

The interface between the CPU (actually, the data link) and the on-site equipment has two facets - hardware and software. The software interface was discussed above for the TWS radar, where it was seen that a 161 bit word must be provided on-site for each target in view every

626 msec. The analysis supporting the selection of these values was also outlined. Similar analyses have been completed for the other radars. Results are summarized in Table III. The indicated word length in each case, if provided at the rate specified, will insure that all accuracy and resolution requirements are satisfied. Therefore, Table III completely defines the software interface between the CPU and the sites.

TABLE III. CPU/RADAR SOFTWARE INTERFACE

RADAR TYPE	WORD LENGTH	UPDATE INTERVAL
TWS	161 bits	626 msec.
CS	171 bits	810 msec.
EW	176 bits	3 sec.

The hardware interface, illustrated in Figure 2, consists of an array of input registers at each site, one for each potential target (see Table I). The existing configuration provides one storage register per target, of word length as indicated in Table III. Thus, each target word can be stored on-site for a period of time corresponding to the update interval for that radar. This provides a tolerance on data computation and data transmission at the CPU. The existing tolerance can easily be increased by expanding the number of input registers, e.g., doubling the number of registers will double the tolerance.

Feedback From Sites To CPU

In establishing simulation requirements, an important objective was to maximize computation performed by the CPU and minimize on-site processing. However, as discussed earlier, the conflicting objective of low data rates between CPU and sites led to a requirement for some on-site computation.

The feedback channel from the site to the CPU provides a means of reducing data rates and on-site computation. If it is known at the CPU how each radar antenna is oriented as a function of time, then targets in the field-of-view of each radar can be identified and only those targets need be transmitted to the respective site. Without feedback it becomes necessary to transmit all targets to each radar, and to then isolate targets within each field-of-view by means of on-site computation. The latter approach would involve higher data rates and require additional on-site computing capacity.

Shown in Table I is the maximum number of aircraft tracks generated for each type of radar.

If the number of aircraft within the field-of-view of the TWS or CS radars exceeds 8 or 4, respectively, then some of the targets must be eliminated from consideration. (It is assumed that the total number of targets generated never exceeds 20, so that a similar condition never arises in the case of the EW radar.) The CPU will eliminate targets as required using an appropriate algorithm, probably to be based on target range. This is an additional computation which must be performed on-site if the feedback channel is eliminated.

Summary

A digital simulation of a defensive test environment is under development. The simulation provides maneuvering target aircraft with ECM jamming for exercises designed to validate command and control systems. The simulation is an integral part of the CCS, which also includes radars, displays, operators, real aircraft with jammers, and a command and control center.

In this continuing program, simulation requirements have been established, modeling is completed or underway, significant interfaces have been defined, and on-site hardware has been developed and demonstrated.

Appendix

Four numerical integration techniques for generating aircraft tracks from equations of motion have been examined, each with a second and fourth-order integration method, and each method applied to two problems whose analytical solution is known. The four techniques selected are all basically predictor-corrector techniques. For the second order method the Nystrom midpoint formula is used as the predictor and the modified Euler formula as the corrector (Reference 1). For the fourth order method the Adams-Bashfourth formula is used as the predictor and the Adams-Moulton formula is used as the corrector (Reference 2). The four techniques employed are:

a) Predict, Correct c times (PC).

b) Predict, Correct c times, Modify (PCM).

c) Predict, Modify, Correct c times (PMC).

d) Predict, Modify, Correct c times, Modify (PMCM).

The first problem investigated was

$$\dot{y}_1 = \frac{1}{y_2} \qquad \dot{y}_2 = -\frac{1}{y_1}$$

$$\dot{y}_1(0) = 1 \qquad \dot{y}_2(0) = 1 \ .$$

The second problem was

$$\dot{y}_1 = y_2 \qquad y_1(0) = 1$$

$$\dot{y}_2 = -(9.25y_1+y_2) \qquad y_2(0) = -1/2 \ .$$

The results for the second-order method can be summarized as follows:

a) PCM is best with PC close second.

b) PMCM is, surprisingly, no better than PCM.

c) My making h, the integration step size, smaller we gain more in accuracy than any change in method produces.

d) Differences between methods are much more pronounced for larger values of h.

For the fourth-order method:

a) PC is best with PCM second.

b) Difference between techniques is not as pronounced as with second order methods.

c) PMCM does relatively better in fourth order than in second order.

d) Making h smaller does not have the impact that it does with the second order method.

In general:

a) Fourth order accuracy increases faster (relative to second order accuracy) with decreasing h.

b) Dividing h by 2 and using second order gives about same, or a slightly better, result than using fourth order with original h.

References

1. M.I. Skolnik, Introduction to Radar Systems. New York: McGraw-Hill, 1962.

2. J.V.Di Franco and W.L. Rubin, Radar Detection. Prentice-Hall: Englewood Cliffs, N.J., 1968.

3. R.W. Hamming, Numerical Methods for Scientists and Engineers. New York: McGraw-Hill, 1962.

4. L. Lapidus and J. Seinfeld, Numerical Solution of Ordinary Differential Equations. New York: Academic Press, 1971.

5. A. Ralston, A First Course in Numerical Analysis. New York: McGraw-Hill, 1965.

6. Study and Design of Signal Injection Systems Range Instrumentation Design Plan, Report No. 86-52313, Westinhgouse Electric Corp., Baltimore, Md., November 19, 1971.

7. Study and Design of Signal Injection Systems, Contract Report: F-8635-72-C-0008, Westinghouse Electric Corp., Baltimore, Md., March 1, 1972.

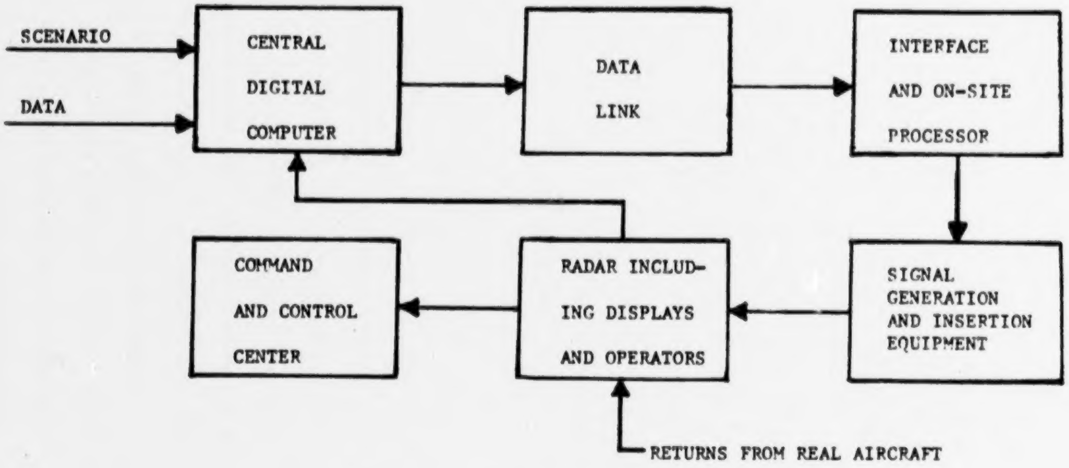

SCENARIO
DATA
CENTRAL DIGITAL COMPUTER
DATA LINK
INTERFACE AND ON-SITE PROCESSOR
SIGNAL GENERATION AND INSERTION EQUIPMENT
RADAR INCLUDING DISPLAYS AND OPERATORS
COMMAND AND CONTROL CENTER
RETURNS FROM REAL AIRCRAFT

FIGURE 1. SIMPLIFIED BLOCK DIAGRAM

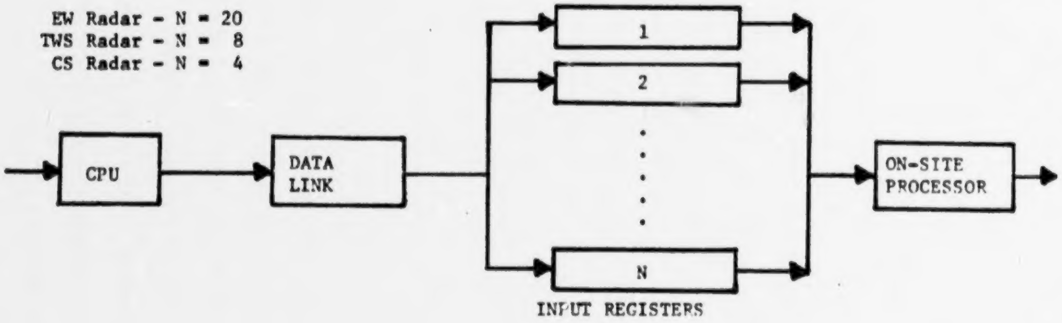

EW Radar - N = 20
TWS Radar - N = 8
CS Radar - N = 4
CPU
DATA LINK
1
2
N
INPUT REGISTERS
ON-SITE PROCESSOR

FIGURE 2. CPU/SITE HARDWARE INTERFACE

COST/RESOURCE MODEL

Betty J. Landstra
Hughes Aircraft Company
Field Service and Support Division

Abstract

The Cost/Resource Model is a simulation model intended for use in management decision-making. The model portrays the impact of proposed new business or changes in existing business on company resources including labor, equipment, facilities, material and money.

INTRODUCTION

A simulation model has been designed for use as a management tool in planning resource requirements and allocations. The model permits determination of the impact on a company of changes in existing contract schedules or product requirements, addition of new contracts to the company's business level, or cancellation of existing contracts.

When a request for a proposal is received, the estimated characteristics of the proposal are examined and its demands aggregated with the requirements of all contracts currently in house to determine what effect this new business will have on company resources. The model can also examine the effect of possible revisions to contracts or proposals currently in house. Individual contracts may be expanded, reduced, deleted, stretched, advanced or delayed. The model then determines the impact on company resources caused by any of these eventualities.

The model permits the forecasting of costs and resources for five years into the future for a medium-sized manufacturing division of a major aerospace company. The model is deterministic and is based on resource data that exists within the company. Considered in the model are all the resources necessary for company operation: labor, material, machines, facilities and money. Another factor considered is other direct costs which includes costs that may not be charged to any of the resources.

A vital aspect of the model is its quick-response capability. Anticipated turn-around time for the model is one day. The model is programmed in Fortran IV in the batch world. Design of the program is modular to permit ease in removing the influence of one or more resources. A given resource may also be isolated simply by zeroing out the inputs for all other resources. Outputs include: headcount per month by type of personnel, total headcount required per month, machine hours required by work center, shortages or surpluses of labor and machines, space requirements per month, resource costs and money requirements. Additional outputs are available as required.

The model was designed specifically for the El Segundo Manufacturing Division of the Hughes Aircraft Company, a manufacturer of complex electronic equipment, but its generality makes it applicable to other companies with a minimum of revision.

STATEMENT OF THE PROBLEM

Management in any company must make decisions affecting operations several years into the future. Some of these decisions concern whether or not to pursue new business opportunities; others concern expansion or contraction of resources such as facilities. To enable management to make these decisions most profitably, adequate information must be accessible concerning resource availability, resource demand and cost of operation.

Under rapidly changing environmental conditions, how is management to determine within a brief period how much manpower will be needed each month for the next five years, how much space these people will occupy, how much machine time will be required and how much money will be needed? How can management adequately ascertain the potential impact of possible new business on manpower, machines, facilities and money. Another problem management faces is revision of contracts currently in house. What would happen to the resource outlook if a contract were stretched out, deliverable items were added or deleted, or a contract were terminated?

Because of this type of question, Hughes El Segundo Manufacturing Division management requested that the tools of management science be used to analyze company operations and develop a model that would provide answers with a minimal lapse of time.

PURPOSE OF THE MODEL

The Cost/Resource Model calculates and portrays the impact of potential new contracts, changes in existing contracts, cancellation of an existing contract, or the effects of changes in resource parameters such as wage rates, holidays, capacity of machines and space, and interest rates. The purpose of the model is to permit management to ask "what if" questions and to see the simulated impact of its decisions so that this information can be incorporated into decision-making strategies.

Use of the model for planning purposes makes it essential that it have a quick-response capability. Anticipated turn-around time is one day or less. This means that answers can be obtained and new alternatives considered in the short time span characteristic of this type of decision-making.

DESCRIPTION OF THE MODEL

The Cost/Resource Model is a deterministic simulation model that forecasts resource requirements and costs by month for five years into the future based on projected resource demands.

The model first aggregates requirements of all contracts currently in house and outstanding proposals which are expected to be captured. A new request for proposal may then be selected for study of its potential impact. Its basic requirements are generated and the resulting demand for resources is superimposed upon the requirements of contracts and proposals currently in house.

Another function of the model is to determine what will happen to resource requirements if a contract or proposal currently in house is terminated or revised. Six options for revision are available; the delivery schedule of end items in one or more contracts may be slipped, advanced, reduced, expanded, deleted or stretched. Under the reduction option, a contract may be reduced by a specified percentage or certain end items may be deleted. In the same way, an expansion to a contract may be effected by adding a percentage or adding a specified number of end items. The effect on resources and costs of changes to parameters such as wage rates, capacity of machines or space, and interest rates may also be determined.

The results of these possible changes are shown in reports showing requirements for manpower, material, space, machines and money. Costs of operation are determined and displayed along with projected revenue and gross earnings. Other additional output options are available at the discretion of the modeler.

Structure of the model is modular to permit revision of the program for estimating one resource without affecting the other resources. A separate module exists for each of the resources: labor, material, machines, facilities and money. Another module is provided for a category of costs called "other direct costs" which are direct costs that cannot be applied to any resource. Separate modules are also provided for spreading the requirements of a new proposal and for revising existing contracts and proposals.

Characteristics of the model are shown in Figure 1. Inputs include requirements for a new proposal, revisions of existing contracts, and parameter changes. Other inputs are data for existing contracts and proposals which involve manpower, machine and material requirements. These requirements are input for each contract;

the model aggregates the requirements, following applicable revisions of the contracts, into total requirements per month.

Shown as circular forms in Figure 1 are the submodels for the resources. These submodels, or modules, summarize the requirements for the resource per month and the costs of the resources per month. Each of these submodels is a program which calls the revision program and the spread program as needed.

A flow diagram of the program relationships is shown in Figure 2. Activity originates in the main program which calls each of the submodels, or programs, in sequence. The first program called is the resource spread program which spreads the resource requirements for the new proposal under consideration. Control then returns to the main program and the first of the resource submodels is called, the material summary model, or program. This program in turn calls the revision program which alters the material requirements per month of a specified contract. For certain types of revisions, it is necessary for the revision program to call the resource spread program to attain the desired revision of resources. Following revision and aggregation of the resulting material requirements over all contracts, control is returned to the main program. Subsequently, each of the resource submodels is called in turn. Requirements for that resource are read from tape or a disk file, revisions are made as required, the resulting total resource requirements per month and resource costs per month are displayed and control is again returned to the main program. Following completion of all programs, the main program calculates total costs and displays gross earnings. Outputs of the model are shown in Figure 1 below the respective submodels.

Submodel Characteristics

Resource Spread Model: The first of the submodels accessed by the main program is the resource spread model, or program. The function of this submodel is to determine resource requirements of a new proposal or contract. This submodel takes as input the total lump-sum requirements for resources for the entire new proposal. It also takes as input the schedule for end item delivery, setback and makespan, the distribution according to which the work is to be spread (eg. according to a learning curve), and the performance factor as shown in Figure 3. The resource spread program spreads the requirements and outputs the direct labor requirements per bid category per month, work center (machine) hours required per month, other direct costs per month and material requirements per month for the new proposal. Output of the resource spread program is stored on disk files and becomes input to the resource submodels.

Material Summary Model: Second of the submodels accessed by the main program is the material summary model Figure 4. The function of this submodel is to determine the requirements (in dollars) for material per month.

Inputs to this model include the material requirements per month for each existing contract and proposal. Also input to this submodel are the monthly material requirements for the new proposal which are an output of the resource spread program. Another input to the model could be a parameter change or a contract revision requirement. The material summary model revises the specified program and aggregates material requirements per month over all contracts and proposals. It then calculates the material burden rate according to the following equation:

$$\text{MBR} \approx \frac{[\frac{a}{12} + \frac{.032}{12}(\text{mat'l req per mo})]\text{x}[1.0 + \frac{.04}{12}]^{12y}}{\text{material required per month}}$$

where: MBR = material burden rate

a = minimum cost for the materiel function

y = number of years since the base year

Material burden rate is the ratio of the cost of acquiring material to the cost of the material itself. Material burden costs are then calculated as follows:

Procurement Costs = (A)(MBR) (Material required per month)

Receiving Inspection Costs = (B)(MBR) (Material required per month)

Receiving Costs = (C)(MBR)(Material required per month)

Finance Costs = (D)(MBR)(Material required per month)

where A, B, C, and D are distribution factors specified for the company.

These functional relationships were derived by conventional management science techniques. The material submodel then calculates the total costs per month for material and material burden and displays this output.

Other Direct Costs Summary Model: The next submodel, Figure 5, considers other direct costs (ODC), which includes those direct costs that cannot be charged to any of the resources. Inputs to the submodel are the ODC per month for the new proposal and the ODC for each existing contract and proposal. As in the material model, requirements for revision of contracts and parameter changes are input to the submodel. The submodel makes the required changes to the existing contracts or proposals and aggregates the ODC requirements per month over all contracts and proposals. This aggregate monthly value is then output to the computer printer.

Machine Summary Model: Machine requirements are considered in the next submodel, Figure 6. Again the inputs include the output of the resource spread submodel (machine requirements per work center per month for the new proposal), machine requirements per work center per month for each existing contract and proposal, and contract revision and parameter change requirements. The machine summary submodel makes the necessary revisions to existing contracts and proposals, then aggregates the machine requirements per work center per month over all contracts and proposals. The monthly requirements are compared with the machine time available (working hours in the month) to provide a statement of the surplus or shortage of machines per month by work center. Whenever a shortage of machine time occurs in a given month, it is necessary to "off load," or subcontract, that portion of the work that the shop cannot handle. The machine summary submodel determines whether subcontracting is required in any work centers for each month. It then determines the number of hours that must be subtracted from labor requirements for that month and calculates the cost of subcontracting. Subcontracting costs involve procurement, receiving inspection, receiving and finance costs similar to material burden costs and are calculated in the same manner as material burden costs. This submodel then takes as input the costs of new machines that it is anticipated will be purchased in each month and the costs of subcontracting that was planned in advance (as distinguished from "off loading"). New machine costs, planned subcontracting costs and "off loading" costs are then aggregated to provide machine costs per month over the five-year time span. Outputs from this model include machine requirements per work center per month, surplus or shortage per work center per month, and total machine costs per month.

Labor Summary Model: Next in the sequence of operation of the Cost/Resource Model is the labor summary submodel, Figure 7. Inputs to this submodel include the manpower requirements per month for the new proposal (output of the resource spread submodel), manpower requirements per month for each existing contract and proposal, and revision data. Manpower requirements are input to this submodel according to the type of personnel involved. The designators for types of labor are called "bid categories." The submodel revises the necessary contracts, then aggregates the manpower requirements (hours per bid category) over all contracts and proposals for each month in the five-year span.

Headcount for each bid category for each month is then obtained by dividing the hours required by the hours available in the month. This headcount is compared with the headcount currently available to determine projected shortage or surplus of manpower in each bid category for each month. Total direct labor requirements per month over all bid categories are then obtained. Next the submodel determines the indirect labor headcount required by the following equation:

$$\text{Indirect labor headcount} = A + C_1(\text{DLH}) - C_2(\text{DLH})^2$$

where: DLH = direct labor headcount
A = indirect labor headcount with zero direct labor headcount
C_1 & C_2 are empirical constants

Indirect labor includes all personnel who are not charging their time directly to a contract. The submodel then adds the direct labor headcount and indirect labor headcount to obtain total headcount per month. Total headcount, direct labor headcount per bid category and shortage or surplus of labor by bid category are printed out by the submodel for each month of the five-year span.

Labor burden costs are then calculated by this submodel. Equations for labor burden costs have been derived through regression analysis on historical data. Included in labor burden are indirect labor costs, fringe benefits, utilities, operating costs other than those charged directly to the contracts, indirect data processing, depreciation, taxes and insurance, rent and leasehold costs, and plant upgrade.

Costs for direct labor per month are then calculated by multiplying the direct labor hours in each bid category by the anticipated average wage rate for that month for that bid category. Total labor costs per month are then obtained by adding direct labor costs and indirect labor costs. An estimated labor burden rate for each month of the five-year time span is calculated by dividing labor burden costs by direct labor costs. This estimated labor burden rate is printed out by the submodel. Other outputs are the components of labor burden, total labor burden and total labor costs for each month.

Facilities Summary Model: The facilities summary submodel is the next program in the sequence. This model takes as input the direct labor headcount per bid category per month and indirect labor headcount per month from the labor summary model as shown in Figure 8. Space requirements per person for each category are input to the model and are multiplied by the appropriate headcount to determine total space requirements per category. A shift factor is included in this calculation to account for the number of people working on the night shifts. These people would occupy the same space occupied by the day shift personnel; consequently, no space is allocated for the second and third shift people.

These space requirements are then aggregated and added to special space requirements, which are input to the model, to obtain total space requirements per month. These requirements are then compared with the anticipated space available, also an input to the model, to determine the shortage or surplus of space for each month over the five-year time span. The submodel then outputs space required for each category per month, total space requirements per month and the shortage or surplus of space over the five-year time span.

Parking space requirements are also calculated by the facilities submodel. The direct labor headcount is modified by the shift factor and then added to the indirect labor headcount. This sum is then divided by a "car pool" factor of 1.2 that accounts for the average number of people riding to work in one car to give the number of parking spaces required per month. This figure is printed out by the model and is compared with the parking spaces available for each month to determine the shortage or surplus of parking spaces.

Another input to the model is planned capital expenditures for new plant for each month of the future five years. These expenditure figures are the facility costs for the time span under consideration and are used in the money summary submodel to calculate total costs.

Money Summary Model: The money summary submodel is the last of the modules in the Cost/Resource Model. Inputs to this submodel are the outputs from the other submodels: the costs per month of labor, machines, material, facilities, and ODC. Labor costs are reduced by the amount of depreciation. This submodel aggregates these costs per month to obtain the net money requirements per month. Revenue per month for the new proposal and for each existing contract and proposal is the input to the submodel. These revenues are summed and compared with the net money requirements to determine whether money must be borrowed each month. The money borrowed is multiplied by the current prime interest rate to determine the cost of money for each month. This cost is added to the net money requirements to obtain the total money requirements per month. Money borrowed, cost of money borrowed and total money requirements per month are printed out by the submodel. The submodel then aggregates over all months in the five-year timespan money requirements and total revenue per month for all contracts. Total money requirements and total revenue for the entire period are then transferred to the main program.

In the main program, depreciation is added to money requirements to obtain total costs. The total costs and total revenue are then compared to obtain gross earnings over the time span under consideration. Total costs and gross earnings are then printed out by the model. These values enable management to compare probable earnings figures for different combinations of contracts and proposals. If the goal is to find the roster of contracts that will provide the greatest earnings, this figure is available from each run of the model with different contracts and proposals included in each run.

USE OF OUTPUTS OF THE MODEL

Predicted resource requirements generated by the model enable management to plan effectively for the future. If manpower available is at a high level at the present time, the model will indicate when a surplus of manpower will occur and the kinds of manpower involved. Generally, contracts in house phase out after a few years and valleys occur in resource requirements. Management must know where those valleys will fall so that new business may be obtained to take up the slack. This model will provide management with this information.

On the other hand, the company may have a large backlog of business with an expected shortage of resources. A request for proposal may come in and management must know whether to bid on it or not. It is important to know over what time span the proposed business would demand resources. If it comes at the peak period, it would be most politic not to submit a bid since the company could not show a capability for performing. It is also necessary for management to know when to start enlarging the work force to accommodate the peak work load. Or, it may be desirable to work overtime for a period rather than to staff up and a few months later have to lay off. It is essential to know how large the gap is between requirements and resource availability to determine whether overtime in lieu of adding workers is feasible.

Space requirements are another important factor in management planning. Will we be short of space for a certain peak period? Will we need to expand space for the long run? Can we change the shift factor and increase the sizes of the night shifts to effect better utilization of space? Management must know when peaks occur, what the trend of space requirements over the long run will be, and when valleys in space utilization occur so as to plan efficiently for adding facilities, planning additional leasing of facilities, or phasing out presently held leaseholds.

The model will provide all of the information required for management to make these types of decisions. The Cost/Resource Model is an effective tool for management in planning resource allocations and determining the most profitable new business to be required.

The model also provides management with a view of what resource allocations would be if a contract currently in house were revised or terminated or if a proposal currently under consideration by a customer were slipped or advanced or changed in scope. Management must be able to react quickly under these circumstances to provide the customer with an estimate of the cost of the proposed revision.

Quick-response is one of the characteristics of the model. Most of the input data are stored on disk files and updated quarterly through the time-sharing terminal. Manpower data are stored on tape which is updated quarterly. Input data which must be prepared at the time the model is run involve requirements for a new proposal, revision requirements for an existing contract or proposal and requirements for planned subcontracting and new machines. Two files are involved with this data. One additional file must be revised when a new proposal or contract revision is under consideration: the revenue file. This state of preparedness and minimization of data preparation at model-run time provide the quick-turnaround time essential to management decision-making.

VEHICLE FOR THE MODEL

The Cost/Resource Model is programmed in Fortran for the batch world of the Honeywell 635 computer. Memory required for the program is 65K. An overlay technique is used to keep program size down as much as possible. Running time for the model is six hundreths of an hour, or three and a half minutes.

Fortran was chosen as the language for the model because the original intention was to operate the model in the timesharing mode and none of the simulation languages were available at that time in the timesharing mode of the 635 computer.

Design, programming and testing of the model have required two years of one person's time. Approximation of the current level of Division activity has also been accomplished within that two-year time span.

DATA COLLECTION

Data for the model were intended to come from sources already available within the company. A previously issued report which has been updated quarterly provides manpower data in the form of hours required per contract, per type of direct personnel, per month, per department, per phase in the contract (implementation, production or production support phases). This report is on tape. Material requirements are obtained from a currently produced report which provides information in the form of material dollars per month per contract for a two-year time span. This report is not presently in a form suitable for direct input to the computer. It must also be extended to a five-year term.

Machine information is obtained from an eight-year forecast of machine requirements. It is anticipated that rather than updating this forecast, a new technique will be used of applying actual times to contract requirements. Space requirements have been analyzed to determine the amount of space occupied by each type of person. Areas not occupied by people, such as warehouses, are input as special space requirements. This information is currently available within the company.

Revenue per month from each of the contracts is obtained from the Finance Department. Other data than those specified here are obtained from various reports and manuals prepared by the company. Included are man/machine ratios, number of machines in each work center, average wage rate for each type of personnel, costs of new facilities per month and subcontracting rates.

Data are stored on tapes or disk files depending on their sources. Disk files are generated in the timesharing mode and are translated to BCD files through the Cardin system, which is the interface between timesharing and the batch world.

MODEL VALIDATION

The model will be validated by using actual data as inputs for a given month or series of months. Total actual requirements will be known for each resource for those months. It will then be possible to verify that the results obtained by the model correlate with the real world.

STATUS OF THE COST/RESOURCE MODEL

Design and programming of the Cost/Resource Model have been completed. All options of the model (with a new proposal, without a new proposal, with each of six revision options) have been tested successfully. One run has been made with partial real data. The model is now running and ready for implementation in a production mode. It is awaiting acquisition of the data necessary for production operation.

GLOSSARY

End Item - The deliverable equipment item specified in the contract

Make Span - The period of time required to manufacture the item

Set Back - The date on which production must begin (obtained by subtracting the makespan from the last end item delivery date)

Gallagher Distributions - A series of six distributions that represent experience in manufacturing effort over the makespan (derived by Paul Gallagher, Hughes Aircraft Company)

Selected Increment distributions - The makespan is divided into twenty equal increments and the proportion of effort for each increment is specified

Turn-Around Time - The period from receipt of new proposal input data and contract revision information to submittal of resource reports to management

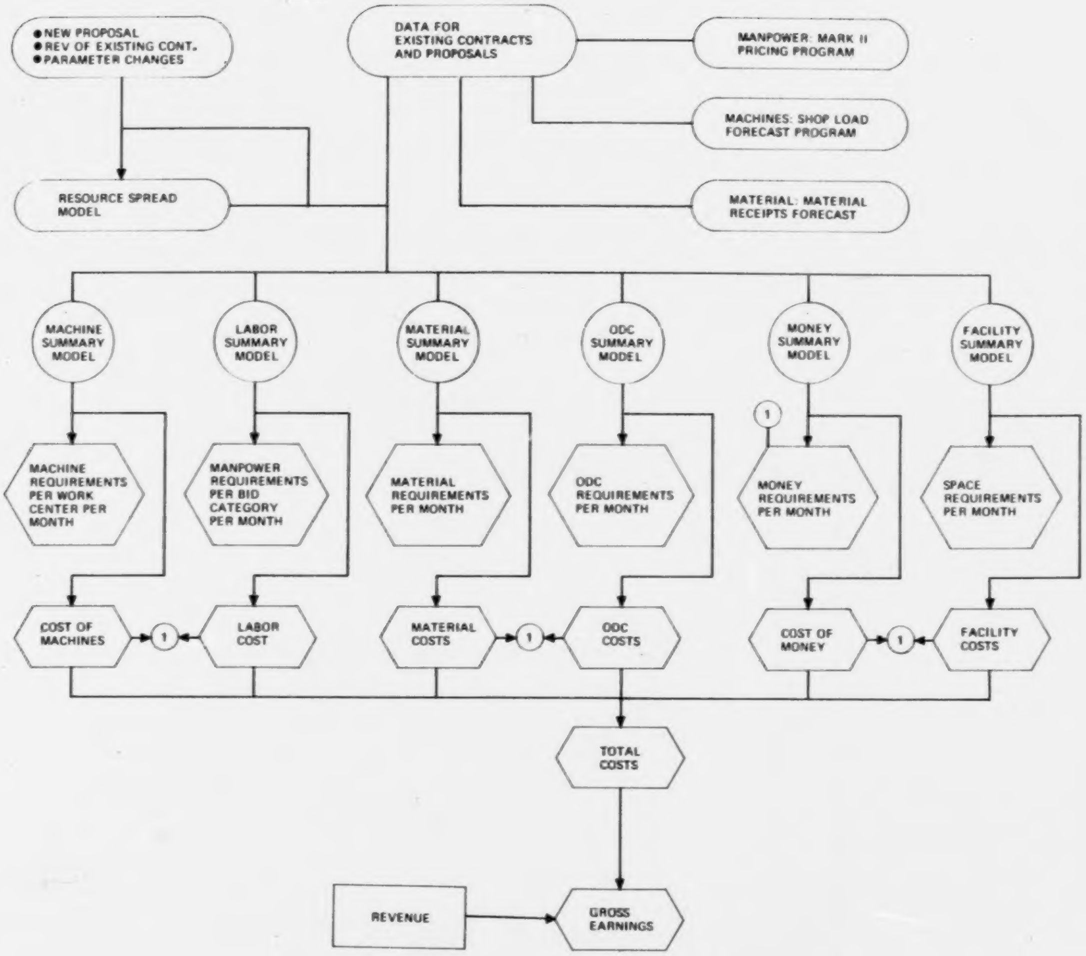

●NEW PROPOSAL
●REV OF EXISTING CONT.
●PARAMETER CHANGES
DATA FOR EXISTING CONTRACTS AND PROPOSALS
MANPOWER: MARK II PRICING PROGRAM
MACHINES: SHOP LOAD FORECAST PROGRAM
RESOURCE SPREAD MODEL
MATERIAL: MATERIAL RECEIPTS FORECAST
MACHINE SUMMARY MODEL
LABOR SUMMARY MODEL
MATERIAL SUMMARY MODEL
ODC SUMMARY MODEL
MONEY SUMMARY MODEL
FACILITY SUMMARY MODEL
1
MACHINE REQUIREMENTS PER WORK CENTER PER MONTH
MANPOWER REQUIREMENTS PER BID CATEGORY PER MONTH
MATERIAL REQUIREMENTS PER MONTH
ODC REQUIREMENTS PER MONTH
MONEY REQUIREMENTS PER MONTH
SPACE REQUIREMENTS PER MONTH
COST OF MACHINES
1
LABOR COST
MATERIAL COSTS
1
ODC COSTS
COST OF MONEY
1
FACILITY COSTS
TOTAL COSTS
REVENUE
GROSS EARNINGS

FIGURE I. COST/RESOURCE MODEL

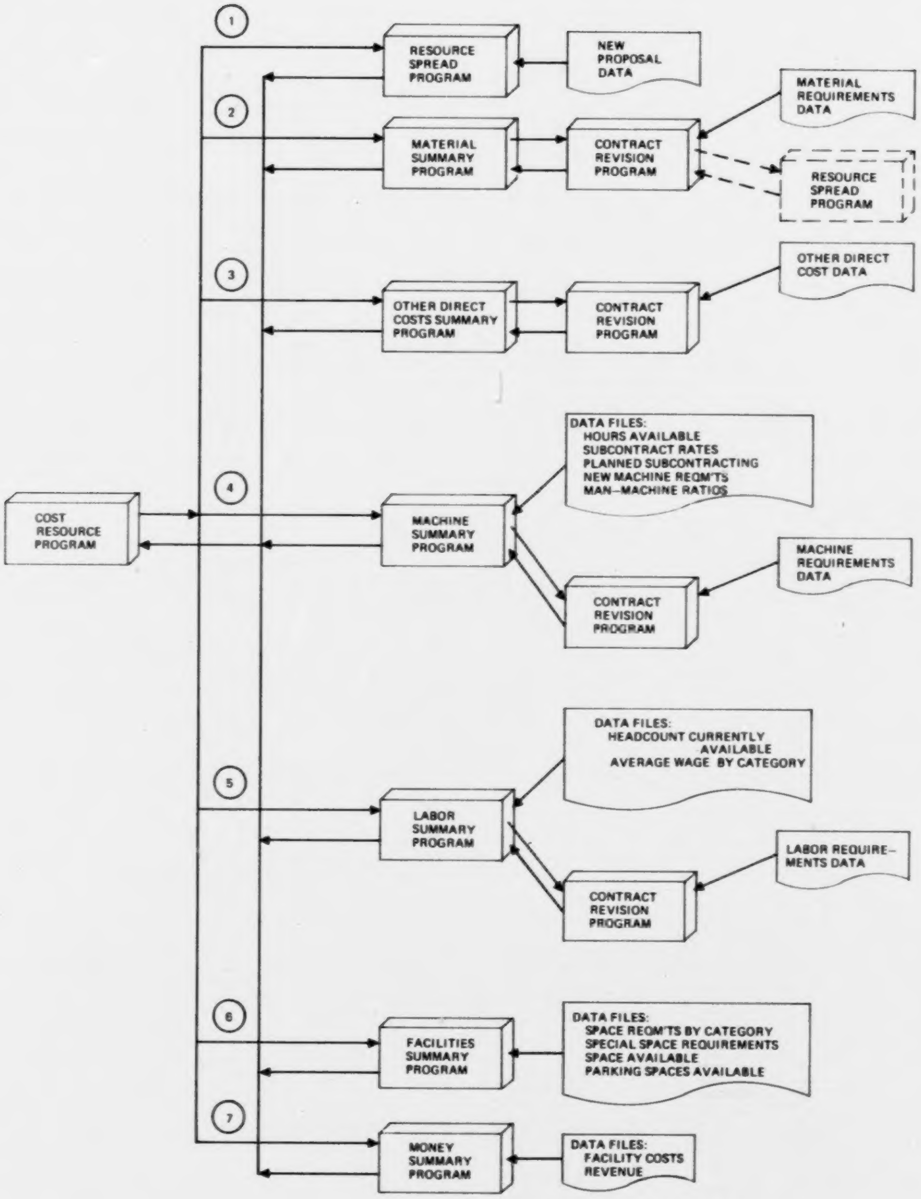

1
RESOURCE SPREAD PROGRAM
NEW PROPOSAL DATA
2
MATERIAL SUMMARY PROGRAM
CONTRACT REVISION PROGRAM
MATERIAL REQUIREMENTS DATA
RESOURCE SPREAD PROGRAM
3
OTHER DIRECT COSTS SUMMARY PROGRAM
CONTRACT REVISION PROGRAM
OTHER DIRECT COST DATA
DATA FILES:
HOURS AVAILABLE
SUBCONTRACT RATES
PLANNED SUBCONTRACTING
NEW MACHINE REQM'TS
MAN–MACHINE RATIOS
4
COST RESOURCE PROGRAM
MACHINE SUMMARY PROGRAM
CONTRACT REVISION PROGRAM
MACHINE REQUIREMENTS DATA
DATA FILES:
HEADCOUNT CURRENTLY AVAILABLE
AVERAGE WAGE BY CATEGORY
5
LABOR SUMMARY PROGRAM
CONTRACT REVISION PROGRAM
LABOR REQUIREMENTS DATA
6
FACILITIES SUMMARY PROGRAM
DATA FILES:
SPACE REQM'TS BY CATEGORY
SPECIAL SPACE REQUIREMENTS
SPACE AVAILABLE
PARKING SPACES AVAILABLE
7
MONEY SUMMARY PROGRAM
DATA FILES:
FACILITY COSTS
REVENUE

FIGURE 2. SEQUENTIAL FLOW DIAGRAM

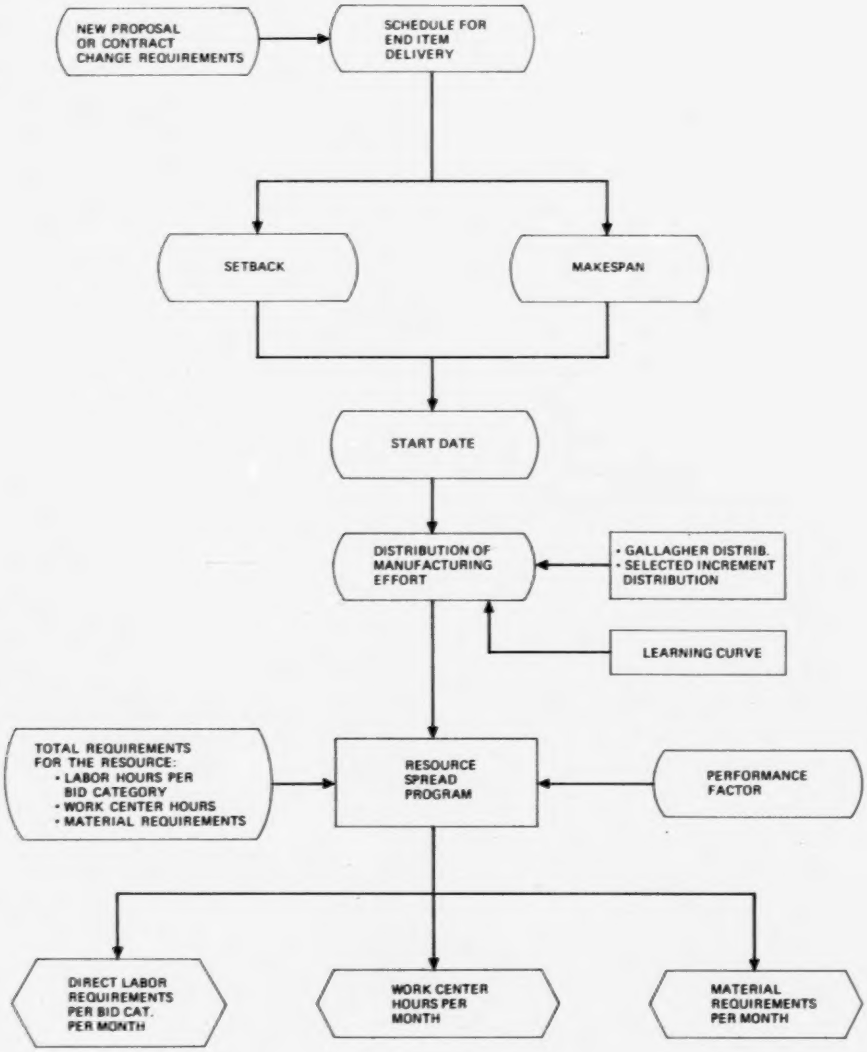

NEW PROPOSAL OR CONTRACT CHANGE REQUIREMENTS
SCHEDULE FOR END ITEM DELIVERY
SETBACK
MAKESPAN
START DATE
DISTRIBUTION OF MANUFACTURING EFFORT
• GALLAGHER DISTRIB.
• SELECTED INCREMENT DISTRIBUTION
LEARNING CURVE
TOTAL REQUIREMENTS FOR THE RESOURCE:
• LABOR HOURS PER BID CATEGORY
• WORK CENTER HOURS
• MATERIAL REQUIREMENTS
RESOURCE SPREAD PROGRAM
PERFORMANCE FACTOR
DIRECT LABOR REQUIREMENTS PER BID CAT. PER MONTH
WORK CENTER HOURS PER MONTH
MATERIAL REQUIREMENTS PER MONTH

FIGURE 3. RESOURCE SPREAD MODEL

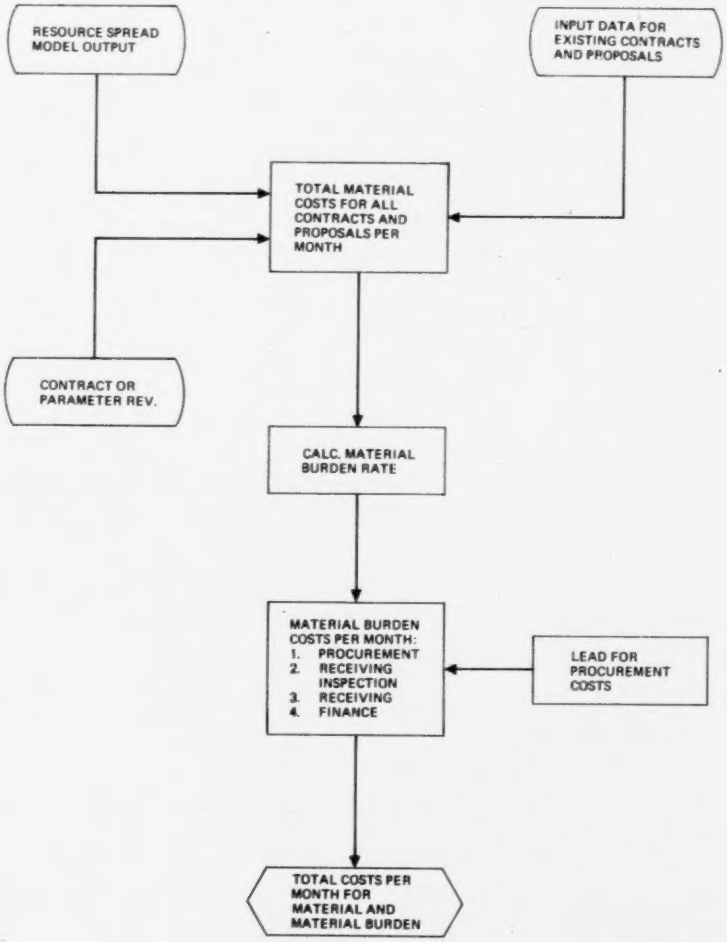

RESOURCE SPREAD MODEL OUTPUT
INPUT DATA FOR EXISTING CONTRACTS AND PROPOSALS
TOTAL MATERIAL COSTS FOR ALL CONTRACTS AND PROPOSALS PER MONTH
CONTRACT OR PARAMETER REV.
CALC. MATERIAL BURDEN RATE
MATERIAL BURDEN COSTS PER MONTH:
1. PROCUREMENT
2. RECEIVING INSPECTION
3. RECEIVING
4. FINANCE
LEAD FOR PROCUREMENT COSTS
TOTAL COSTS PER MONTH FOR MATERIAL AND MATERIAL BURDEN

FIGURE 4. MATERIAL SUMMARY MODEL

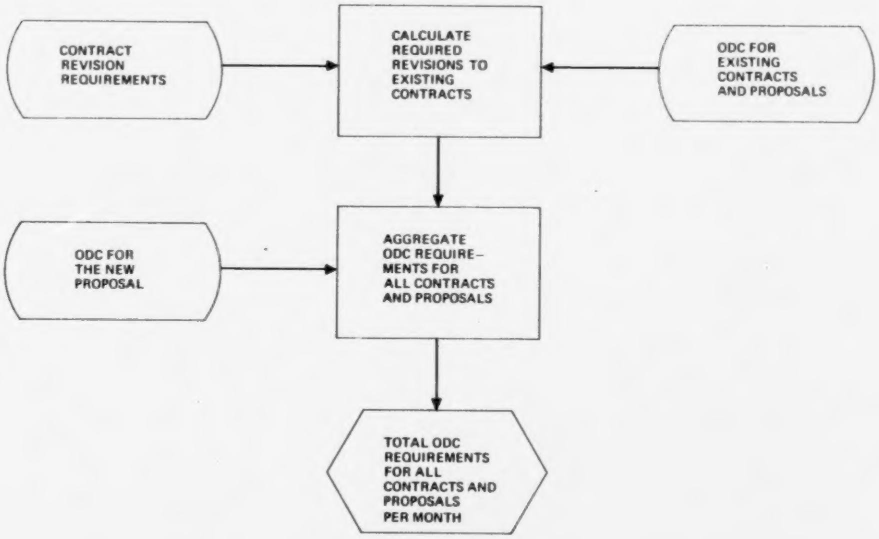

CONTRACT REVISION REQUIREMENTS
CALCULATE REQUIRED REVISIONS TO EXISTING CONTRACTS
ODC FOR EXISTING CONTRACTS AND PROPOSALS
ODC FOR THE NEW PROPOSAL
AGGREGATE ODC REQUIREMENTS FOR ALL CONTRACTS AND PROPOSALS
TOTAL ODC REQUIREMENTS FOR ALL CONTRACTS AND PROPOSALS PER MONTH

FIGURE 5. ODC (OTHER DIRECT COSTS) SUMMARY MODEL

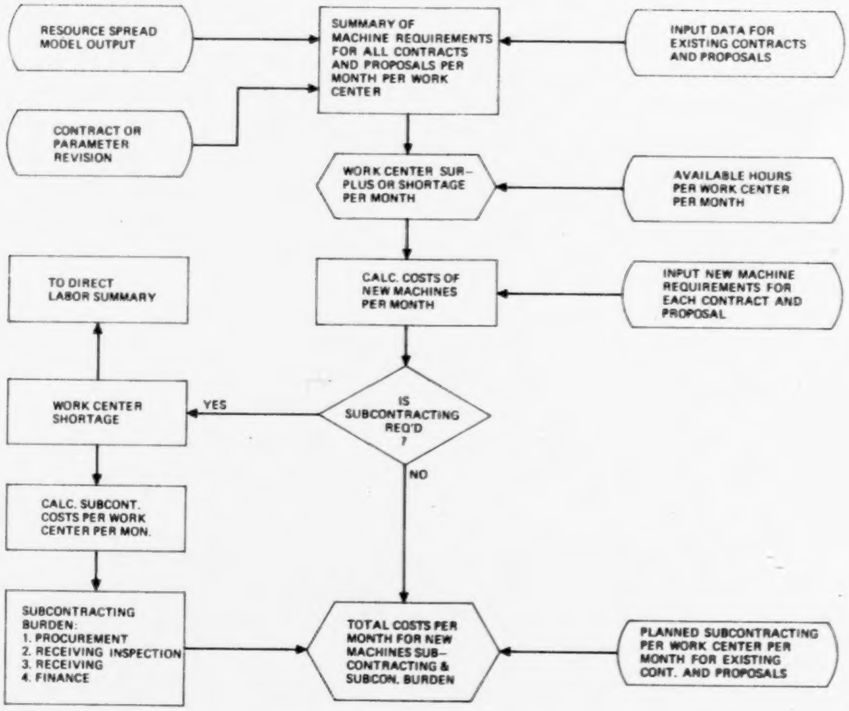

RESOURCE SPREAD MODEL OUTPUT
SUMMARY OF MACHINE REQUIREMENTS FOR ALL CONTRACTS AND PROPOSALS PER MONTH PER WORK CENTER
INPUT DATA FOR EXISTING CONTRACTS AND PROPOSALS
CONTRACT OR PARAMETER REVISION
WORK CENTER SURPLUS OR SHORTAGE PER MONTH
AVAILABLE HOURS PER WORK CENTER PER MONTH
TO DIRECT LABOR SUMMARY
CALC. COSTS OF NEW MACHINES PER MONTH
INPUT NEW MACHINE REQUIREMENTS FOR EACH CONTRACT AND PROPOSAL
WORK CENTER SHORTAGE
YES
IS SUBCONTRACTING REQ'D ?
NO
CALC. SUBCONT. COSTS PER WORK CENTER PER MON.
SUBCONTRACTING BURDEN:
1. PROCUREMENT
2. RECEIVING INSPECTION
3. RECEIVING
4. FINANCE
TOTAL COSTS PER MONTH FOR NEW MACHINES SUB-CONTRACTING & SUBCON. BURDEN
PLANNED SUBCONTRACTING PER WORK CENTER PER MONTH FOR EXISTING CONT. AND PROPOSALS

FIGURE 6. MACHINE SUMMARY MODEL

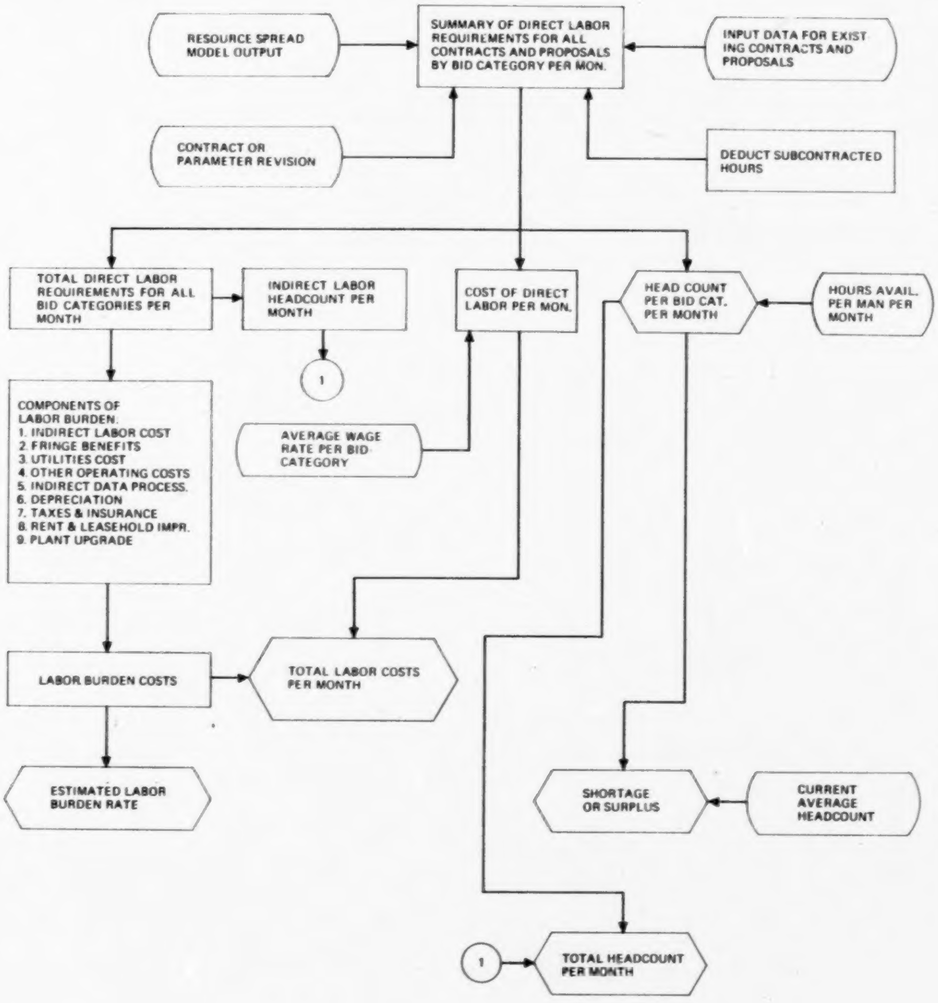

RESOURCE SPREAD MODEL OUTPUT
SUMMARY OF DIRECT LABOR REQUIREMENTS FOR ALL CONTRACTS AND PROPOSALS BY BID CATEGORY PER MON.
INPUT DATA FOR EXISTING CONTRACTS AND PROPOSALS
CONTRACT OR PARAMETER REVISION
DEDUCT SUBCONTRACTED HOURS
TOTAL DIRECT LABOR REQUIREMENTS FOR ALL BID CATEGORIES PER MONTH
INDIRECT LABOR HEADCOUNT PER MONTH
COST OF DIRECT LABOR PER MON.
HEAD COUNT PER BID CAT. PER MONTH
HOURS AVAIL. PER MAN PER MONTH
1
COMPONENTS OF LABOR BURDEN.
1. INDIRECT LABOR COST
2. FRINGE BENEFITS
3. UTILITIES COST
4. OTHER OPERATING COSTS
5. INDIRECT DATA PROCESS.
6. DEPRECIATION
7. TAXES & INSURANCE
8. RENT & LEASEHOLD IMPR.
9. PLANT UPGRADE
AVERAGE WAGE RATE PER BID CATEGORY
LABOR BURDEN COSTS
TOTAL LABOR COSTS PER MONTH
ESTIMATED LABOR BURDEN RATE
SHORTAGE OR SURPLUS
CURRENT AVERAGE HEADCOUNT
1
TOTAL HEADCOUNT PER MONTH

FIGURE 7. LABOR SUMMARY MODEL

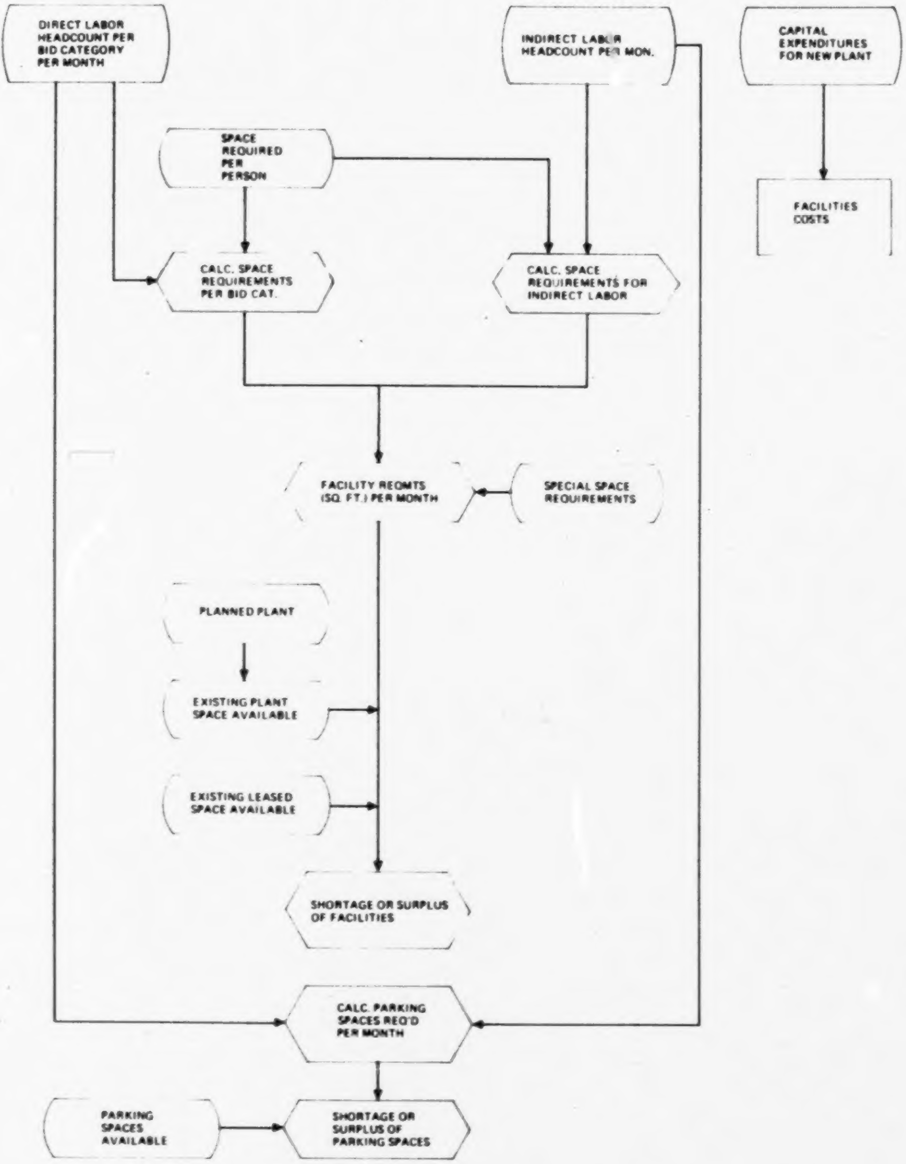
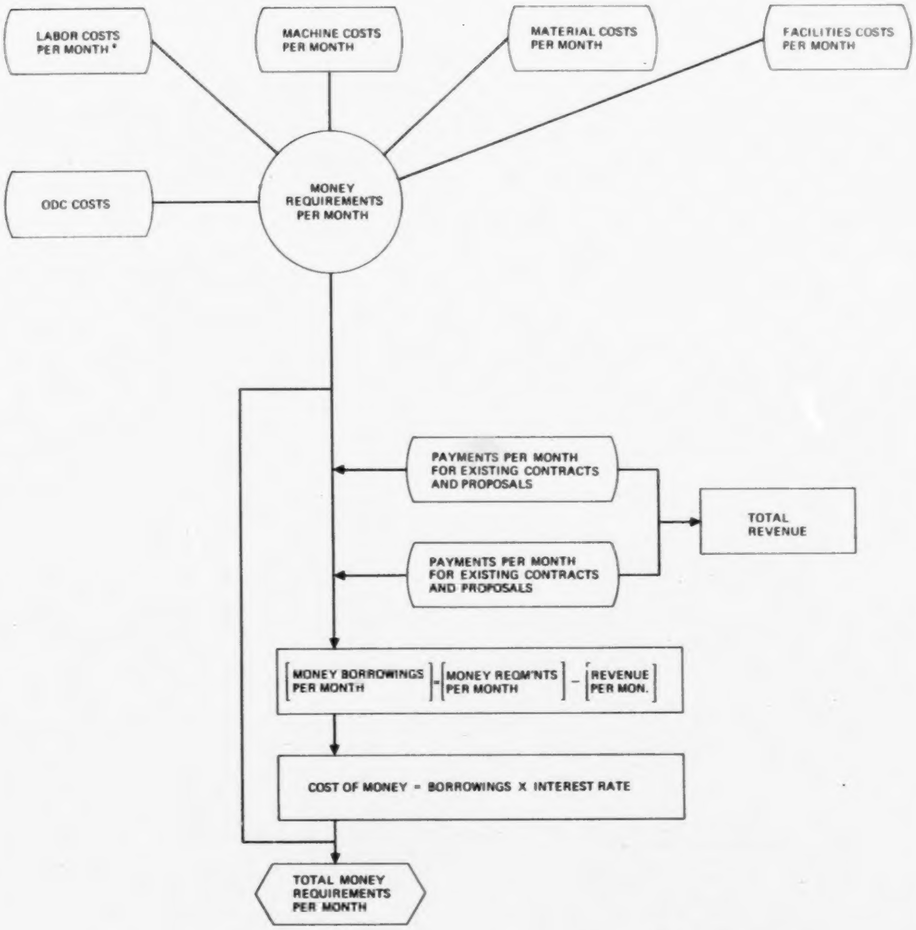

DIRECT LABOR HEADCOUNT PER BID CATEGORY PER MONTH
INDIRECT LABOR HEADCOUNT PER MON.
CAPITAL EXPENDITURES FOR NEW PLANT
SPACE REQUIRED PER PERSON
FACILITIES COSTS
CALC. SPACE REQUIREMENTS PER BID CAT.
CALC. SPACE REQUIREMENTS FOR INDIRECT LABOR
FACILITY REQMTS (SQ. FT.) PER MONTH
SPECIAL SPACE REQUIREMENTS
PLANNED PLANT
EXISTING PLANT SPACE AVAILABLE
EXISTING LEASED SPACE AVAILABLE
SHORTAGE OR SURPLUS OF FACILITIES
CALC. PARKING SPACES REQ'D PER MONTH
PARKING SPACES AVAILABLE
SHORTAGE OR SURPLUS OF PARKING SPACES

FIGURE 8. FACILITIES SUMMARY MODEL


LABOR COSTS PER MONTH *
MACHINE COSTS PER MONTH
MATERIAL COSTS PER MONTH
FACILITIES COSTS PER MONTH
ODC COSTS
MONEY REQUIREMENTS PER MONTH
PAYMENTS PER MONTH FOR EXISTING CONTRACTS AND PROPOSALS
TOTAL REVENUE
PAYMENTS PER MONTH FOR EXISTING CONTRACTS AND PROPOSALS
[MONEY BORROWINGS PER MONTH] = [MONEY REQM'NTS PER MONTH] − [REVENUE PER MON.]
COST OF MONEY = BORROWINGS X INTEREST RATE
TOTAL MONEY REQUIREMENTS PER MONTH

* LESS DEPRECIATION

FIGURE 9. MONEY SUMMARY MODEL

MAFLOS-A GENERALIZED MANUFACTURING SYSTEM SIMULATOR

Kazuyuki Mitome
Susumu Tsuhara
Susumu Seki
Ken'ichi Isoda

Central Research Laboratory, Hitachi, Ltd.
Kokubunji, Tokyo, JAPAN

SUMMARY

Manufacturing systems are very complicated, so it is very difficult to grasp the actual behaviour of the manufacturing processes in detail. Even if we obtain a new idea of the scheduling algorithm or the process layout, it takes a long time until the idea is put into practical use, because it is difficult to prove the actual validity of the idea.

The simulation technique is valid to solve this bottleneck. However, the conventional simulators lack the ability to simultaneously simulate the material flows and the control systems' behaviour.

In this papers, the authors analyze the functions of manufacturing system in the following factors:
(a) equipment layout (b) control system,
(c) material, (d) operation.
According to the analysis, we present a new type simulator which is named "MAFLOS".

MAFLOS is characterized by the following features.
(a) MAFLOS has seven kinds of unit-element to describe the equipment layout of the manufacturing system. The simulation model is therefore generated by combining these unit-element.
(b) MAFLOS can simultaneously simulate the material flow and the control system behaviour.

1. INTRODUCTION

Manufacturing systems are very complicated, so it is very difficult to grasp the actual behaviour of the manufacturing processes in detail. Even if we obtain a new idea of the scheduling algorithm or the process layout, it takes a long time until the idea is put into practical use, because it is difficult to prove the actual validity of the idea. Simulation[2] is one of the best methods to examine the validity of such idea.

Conventional simulators such as GPSS, SIMSCRIPT etc., which are generally used to solve queueing problems[3][4], lack the ability to simulate the control systems' behaviour. Hence, the conventional simulators are not convenient for simultaneously investigating the material flow and control system.

In this paper, the authors will present MAFLOS (**Ma**terial **Fl**ow **S**imulator: a new type simulator for manufactiring systems). MAFLOS has the function to simulate the control systems' behaviour, the process layout and the material flow.

2. ACTIVITY OF MANUFACTURING SYSTEM

We now consider the activity of the manufacturing system shown in Figure 1 as an example. In this system, the materials are manufactured in the sequence which follows:

(1) The materials which arrived from external system are loaded in the "container."
(2) The containers are transfered to the central "storage" by the "handling machines" and the "conveyor."
(3) The central handling machine transfers the containers from the central storage to the local storages which are located beside the "production machines." The containers are transfered according to the sequence of the work schedule for the production machine.
(4) The handling machine selects a material which is demanded by the production machine, from the container.
(5) The selected material is put on the production machine.
(6) The production machine works on the material.
(7) The handling machine returns the worked material to the container.
(8) The central handling machine sends back the container to the central storage. The container stays at the central storage until another production machine requests it.
(9) The materials flow through production machines according to the manufacturing sequence individually given. The manufacturing sequence for each material is predetermined.
(10) The finished materials are taken away by the conveyor and the handling machines.

following two sets of data:
(a) The state of materials.
(b) The state of unit-equipment.
For example, the manufacturing time at the production machine is calculated according to time standard in data set (a) and probabilistic disturbance (if necessary). The carrying time of a handling machine is calculated according to the present position of the handling machine, in data set (b), and the present position of material, in data set (a), and position where the material should be transfered, in data set (b).

3.4.2 Confirming function for the starting conditions

Proceeding the start of an operation, the confirmation of the starting condition must be done. As for the starting condition, there are two cases.
(a) Synchronization of materials: Before the start of assembling operations, all parts that are needed to assemble must be put together.
(b) Propriety of the manufacturing sequence: Before the start of manufacturing operations, whether the operation is on the correct manufacturing sequence or not must be examined.

4. THE STATE OF THE MANUFACTURING SYSTEM

4.1 The state description of the manufacturing system

The state of manufacturing system is described by the state of the materials and the state of the unit-equipment. MAFLOS has the function to describe the following state of the manufacturing system.
(1) The state of materials.
  (a) The present position.
  (b) The progress on the manufacturing sequence.
  (c) The classification of material quality: a good material or no good material.
  (d) The classification of materials in waiting state: the material waiting the operation in the unit-equipment or the material waiting the scheduling in the control system.
(2) The state of the unit-equipment.
  (a) The state of operation of the unit-equipment.
  (b) The progress on the work schedule.
  (c) The name of the material on the unit-equipment.
  (d) The present location of the unit-equipment.

The transition diagram in the state of the operation for the production machine is shown in Figure 4.

The items deciding the state of each unit-equipment are shown in Table 2. A sign "O" in Table 2 denotes that the state is necessary for defining the state of unit-equipment.

4.2 Events inducing the state change

The state of the materials and the unit-equipment in the MAFLOS is changed by the following events.
(a) The start and the end of the working hours.
(b) The operation completion of the unit-equipment (handling machine, production machine, conveyor).
(c) The occurrence of the disturbance (the troubles at the unit-equipment, the manufacturing failure etc.).

5. THE STRUCTURE OF MAFLOS

5.1 The discription of the model of the manufacturing system

The simulation model is generated by feeding the information shown in Fig. 5, to MAFLOS. Shown in Table 3 and Table 4 are the partial input data for the model shown in Figure 1.

5.2 Output (simulation reports)

MAFLOS prepares the following output items for simulation reports.
(1) The interim simulation reports at arbitrary time.
  (a) The state of materials.
  (b) The utilization rate of storages.
An example is shown in Table 5.
(2) The final simulation reports.
  (a) The lead time of each product.
  (b) The rate of operation of the unit-equipment (the handling machines, the production machines, the conveyors).
  (c) The work schedules of the unit-equipment (if necessary).
An example is shown in Table 6.

The output items can be easily extended.

5.3 The program structure of MAFLOS

The whole structure of program is shown in Figure 6. The simulating program for the operation of H/M, P/M and C/V is controlled by the time advance routine and renews the state of the unit-equipment and the state of the materials. If it is necessary to run the scheduling programs, the supervisory routine initiates the scheduling programs. The scheduling programs regenerate the work schedule in accordance with the new situation.

6. CONCLUSIONS

The fundamental factors which characterize the manufacturing systems are the following four items:
(a) equipment layout.

3. DESCRIPTION OF MANUFACTURING SYSTEM IN MAFLOS

In MAFLOS, we classified the factors which characterize the manufacturing systems' behaviour into the following four categories.
(a) equipment layout
(b) control system
(c) materials
(d) operations
We will describe each category in the following sections.

3.1 Equipment layout

The equipment layout is defined by the following two items.
(a) The functions of each unit-equipment.
(b) The connections between unit-equipment.

3.1.1 Classification of unit-equipment

Unit-equipment is classified into the following seven classes according to their functions.
(a) Production machine (P/M): the material are manufactured by the production machines, taking the machine tools as an example.
(b) Handling machine (H/M): a handling machine transfers the materials from one place to another. The crane which moves along the path is an example. Workers carrying the material are also regarded as the handling machines.
(c) Connector (C/N): a connector is a simplified handling machine.
(d) Rail (R/L): a rail is a path along which the handling machine is guided.
(e) Conveyor (C/V): a conveyor is a carrying machine as the belt conveyor or the overhead conveyor.
(f) Conveyor Guide (C/G): a conveyor guide is a path along which the conveyor is guided.
(g) Storage (S/G): a storage is a unit-equipment which stocks the materials.

3.1.2 Expression for the unit-equipment and their connection

In the simulator-MAFLOS, the unit-equipment represents as shown in Table 1. The main parameters which specify the ability of the unit-equipment are also shown in Table 1.

The connection between unit-equipment A and B is defined by the pair (A, B). If A and/or B have some substate, the pair must include the index of substate as position or coordinate.

The manufacturing system which is shown in Figure 1 is expressed as shown in Figure 2 by using Table 1.

3.2 Control system

MAFLOS has the functions to simulate the control system with respect to
(a) Material flow detection
(b) Scheduling

3.2.1 Material flow detecting function for synchronizing the material flow and the control system.

Material flow detecting function is important for obtaining the information of material flows. When the material passes through the predetermined point in the layout of the manufacturing system, the information about the material flow is transmitted to the control system. The material flow detecting function in MAFLOS are classified into the following two items.
(a) Material flow detecting function for incoming materials.
(b) Material flow detecting function for material being removed.

3.2.2 Scheduling function

The production machines work on the materials according to the work schedule. The work schedule is regenerated according to the scheduling algorithm suited for the controlled manufacturing process. Scheduling programs can be incorporated into MAFLOS and the timing the scheduling in MAFLOS can be selected in the following three manners:
(a) When the material flow is detected by the material flow detecting function.
(b) When workers cannot maintain the work schedule previously given.
(c) When a given period of time passes.

3.3 Material

MAFLOS can accept the following information about the materials in relation to the production machine and the handling machine.
(1) As for the production machine.
  (a) Information of the product structures as shown in Figure 3.
  (b) Information of the manufacturing sequence for each product.
  (c) Information of production planning.
(2) As for the handling machine.
  (a) Information of the carrying unit, for example container or pallet, for each operation of the handling machine. However, the quantity to be carrying can be changed by scheduling within the carrying unit.

3.4 Operation

In MAFLOS, the operation of each unit-equipment is evaluated by the time elasping in the operation, therefore the function to calculate the operation time is required. The confirming function (the starting conditions are complete or not) is also required in order to evaluate the scheduling results.

3.4.1 Calculation of the operation time

The amount of time for each operation of a unit-equipment is calculated from the

(b) control system.
(c) material.
(d) operation.

The authors analyzed these fundamental factors in detail, and consequently proposed a new type simulator MAFLOS.

MAFLOS is characterized by the following features:

(a) MAFLOS has seven kinds of unit-elements to describe the equipment layout of the manufacturing system. The simulation model is therefore generated by combining these unit-elements.
(b) MAFLOS can simultaneously simulate the material flow and the control system behaviour.
(c) MAFLOS has "material flow detecting function" in order to synchronize the simulation of the material flow and the simulation of the control system behaviour.
(d) MAFLOS has "confirming function" in order to evaluate the scheduling results.

MAFLOS is suited for the design of the layout and/or the control system. This simulator is especially suited for the design of total manufacturing system including the control system and the layout.

REFERENCES

1) K. Mitome and S. Mitsumori: Schedule Control of Multi-Commodity Mass-Production Systems, 1972 Conference on Islands of Applications, IEEE (Computer Society)
2) Tocker, K. D.: The Art of Simulation, D. Van Nostrand Co. Inc., Princeton, N. J., (1963)
3) J. H. Mize and J. G. Cox: Essentials of Simulation, Prentice-Hall, Inc., (1968)
4) A. R. Pai and K. L. McRoberts: Simulation Research in Interchangeable Part Manufacturing, Management Science, 17-12 B732-B742, (1972)

BIOGRAPHIES

Kazuyuki Mitome is a Researcher in the Central Research Laboratory of Hitachi, Ltd. He has been engaged in research for production control. He received the B.S. degree in Mechanical Engineering from the University of Kanagawa in 1965. He is a member of the Japan Society of Mechanical Engineers, the Institute of Electrical Engineers of Japan and the Society of Instrument and Control Engineers.

Susumu Tsuhara is a Researcher in C.R.L. of Hitachi. He has been engaged in research for production control. He received the B.S. degree in Electrical Engineering from the Kyushu Institute of Technology in 1970. He is a member of the Operations Research Society of Japan, the Institute of Electronics and Communication Engineers of Japan.

Susumu Seki is a Senior Researcher in C.R.L. of Hitachi. He is a member of O.R.S.J. and I.E.C.E.J.

Ken'ichi Isoda is a Senior Researcher in C.R.L. of Hitachi. He is a member of I.E.E.J. and S.I.C.E.

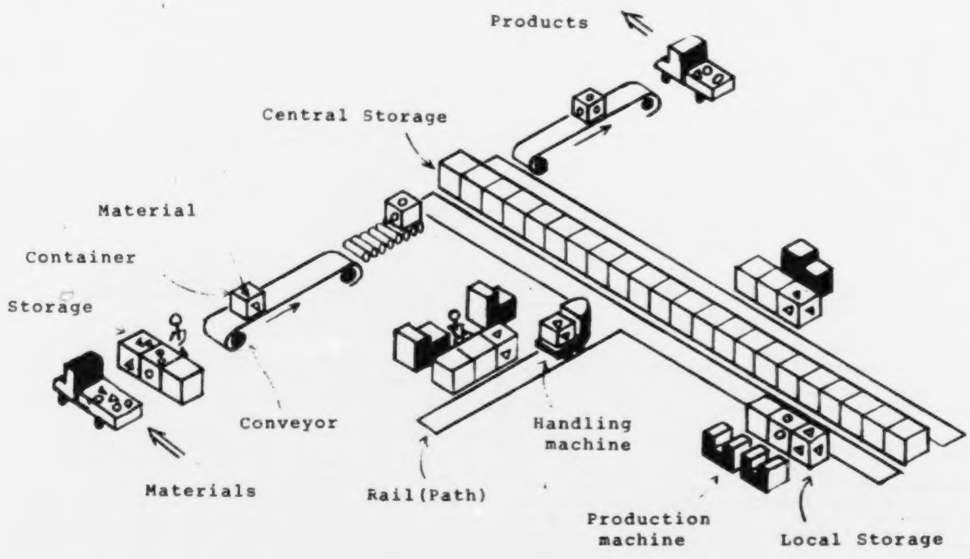


Fig. 1 Manufacturing system

Table 1 Symbols for unit-equipment

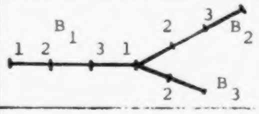
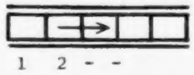
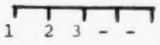
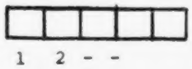
Unit-equipment	Symbol	Main parameter
Production machine	■	Manufacturing speed
Handling machine	○	Transfer speed Pick up and take down time
Connector	△	Rail NO.
Rail		Coordinate(Branch NO.,X,Z)
Conveyor		Conveyor speed Conveyor length Conveyor guide NO.
Conveyor guide		Coordinate (X)
Storage		Capacity (X,Y,Z)

Table 2 Items deciding the state of each unit-equipment

ITEM \ UNIT-EQUIPMENT	PRODUCTION MACHINE	HANDLING MACHINE	CONNECTOR	RAIL	CONVEYOR	CONVEYOR GUIDE	STORAGE
STATE OF MANUFACTURING	○	○	○	·	○	·	·
PROGRESS ON WORK SCHEDULE	○	○	·	·	·	·	·
MATERIAL NAME ON UNIT-EQUIPMENT	○	○	○	·	○	·	○
POSITION OF UNIT-EQUIPMENT	·	○	·	○	○	○	·

EXPLANATION: A SIGN "O" DENOTES THAT THE ITEM IS NECESSARY FOR DEFINEING THE STATE OF THE UNIT-EQUIPMENT

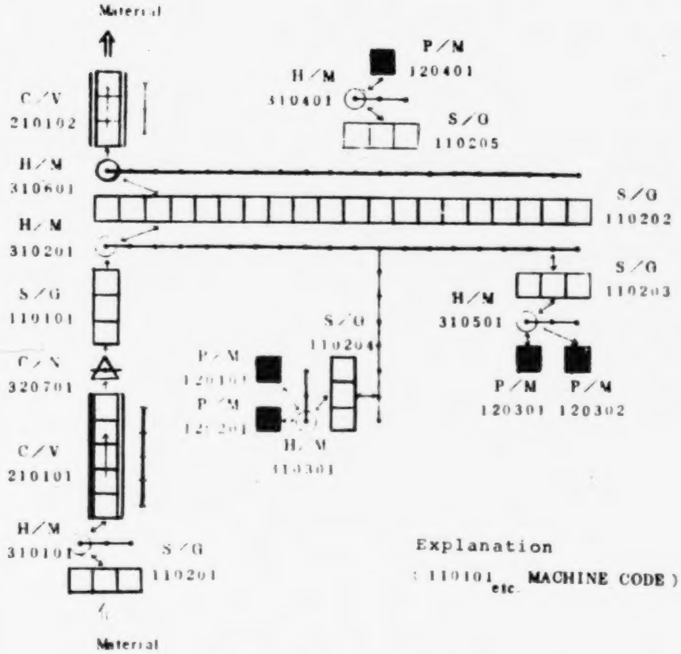

Material
P/M
120401
H/M
310401
S/G
110205
C/V
210102
H/M
310601
S/G
110202
H/M
310201
S/G
110203
S/G
110101
H/M
310501
S/G
110204
C/N
320701
P/M
120101
P/M
120201
P/M
120301
P/M
120302
C/V
210101
H/M
310301
H/M
310101
S/G
110201
Explanation
(110101 etc. MACHINE CODE)
Material

Fig. 2 Manufacturing system described by MAFLOS' symbols

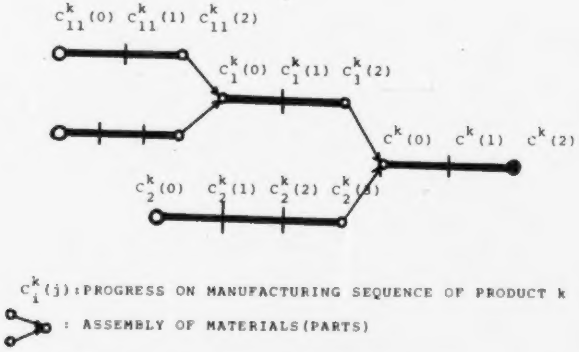

$c^k_{11}(0)$ $c^k_{11}(1)$ $c^k_{11}(2)$
$c^k_1(0)$ $c^k_1(1)$ $c^k_1(2)$
$c^k(0)$ $c^k(1)$ $c^k(2)$
$c^k_2(0)$ $c^k_2(1)$ $c^k_2(2)$ $c^k_2(3)$
$c^k_i(j)$: PROGRESS ON MANUFACTURING SEQUENCE OF PRODUCT k
: ASSEMBLY OF MATERIALS (PARTS)

Fig. 3 Product structure

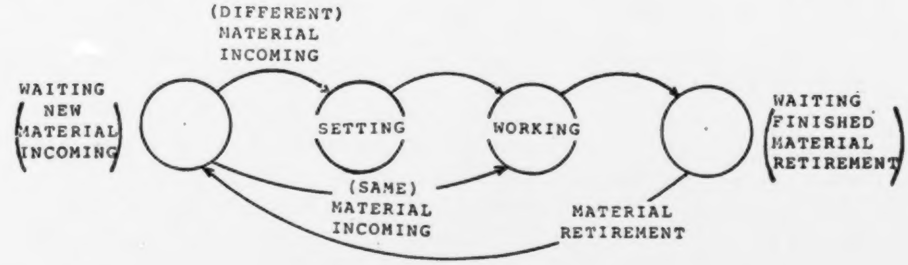
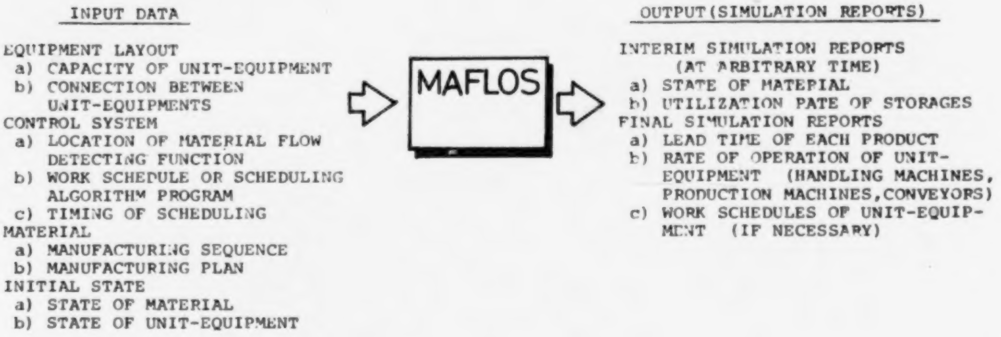

(DIFFERENT) MATERIAL INCOMING
WAITING NEW MATERIAL INCOMING
SETTING
WORKING
WAITING FINISHED MATERIAL RETIREMENT
(SAME) MATERIAL INCOMING
MATERIAL RETIREMENT

Fig. 4 Transition diagram in the state of the operation for the production machine



INPUT DATA

EQUIPMENT LAYOUT
- a) CAPACITY OF UNIT-EQUIPMENT
- b) CONNECTION BETWEEN UNIT-EQUIPMENTS

CONTROL SYSTEM
- a) LOCATION OF MATERIAL FLOW DETECTING FUNCTION
- b) WORK SCHEDULE OR SCHEDULING ALGORITHM PROGRAM
- c) TIMING OF SCHEDULING

MATERIAL
- a) MANUFACTURING SEQUENCE
- b) MANUFACTURING PLAN

INITIAL STATE
- a) STATE OF MATERIAL
- b) STATE OF UNIT-EQUIPMENT

OUTPUT(SIMULATION REPORTS)

INTERIM SIMULATION REPORTS (AT ARBITRARY TIME)
- a) STATE OF MATERIAL
- b) UTILIZATION RATE OF STORAGES

FINAL SIMULATION REPORTS
- a) LEAD TIME OF EACH PRODUCT
- b) RATE OF OPERATION OF UNIT-EQUIPMENT (HANDLING MACHINES, PRODUCTION MACHINES, CONVEYORS)
- c) WORK SCHEDULES OF UNIT-EQUIPMENT (IF NECESSARY)

Fig. 5 Input data and simulation reports

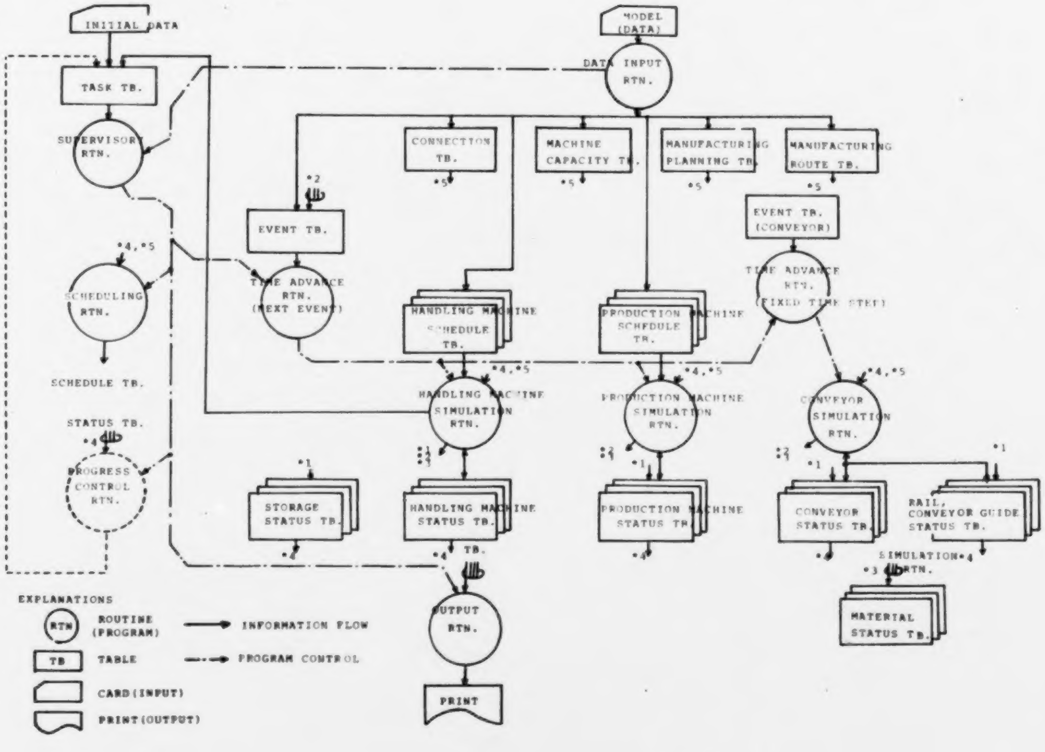
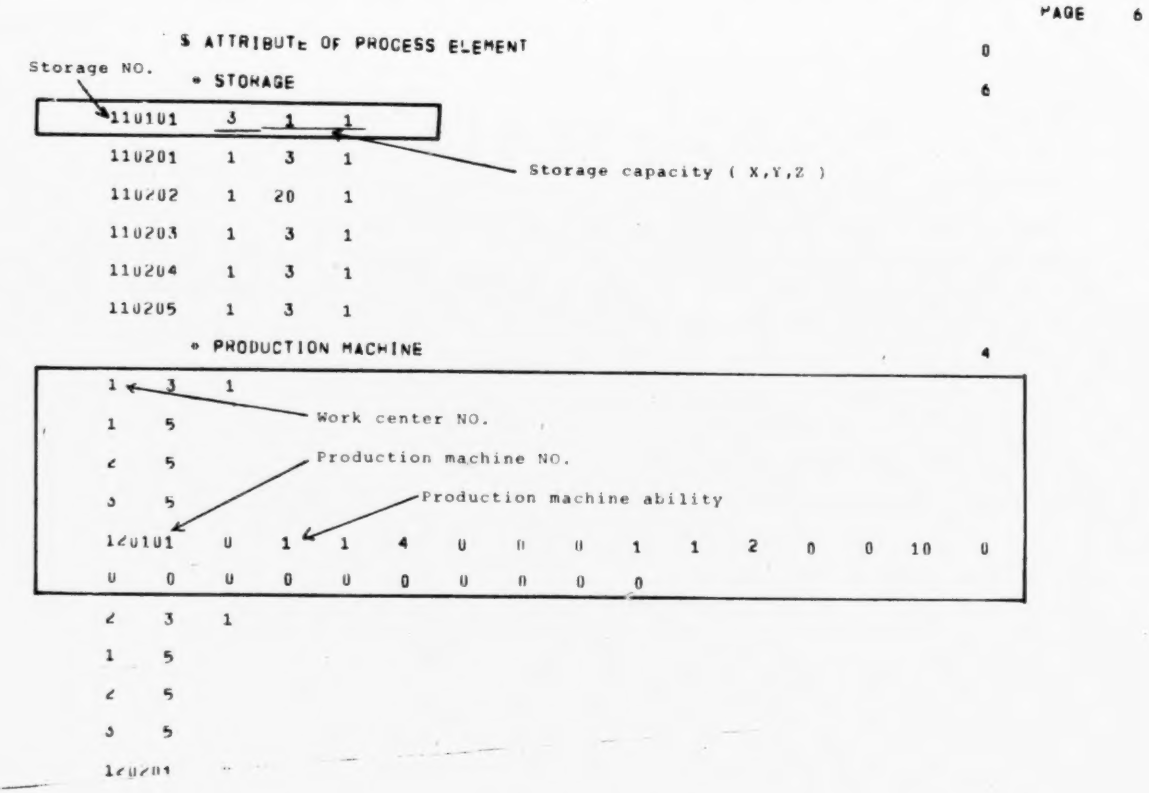

INITIAL DATA
MODEL (DATA)
DATA INPUT RTN.
TASK TB.
SUPERVISORY RTN.
CONNECTION TB.
MACHINE CAPACITY TB.
MANUFACTURING PLANNING TB.
MANUFACTURING ROUTE TB.
EVENT TB.
EVENT TB. (CONVEYOR)
SCHEDULING RTN.
TIME ADVANCE RTN. (NEXT EVENT)
HANDLING MACHINE SCHEDULE TB.
PRODUCTION MACHINE SCHEDULE TB.
TIME ADVANCE RTN. (FIXED TIME STEP)
SCHEDULE TB.
STATUS TB.
HANDLING MACHINE SIMULATION RTN.
PRODUCTION MACHINE SIMULATION RTN.
CONVEYOR SIMULATION RTN.
PROGRESS CONTROL RTN.
STORAGE STATUS TB.
HANDLING MACHINE STATUS TB.
PRODUCTION MACHINE STATUS TB.
CONVEYOR STATUS TB.
RAIL, CONVEYOR GUIDE STATUS TB.
SIMULATION RTN.
MATERIAL STATUS TB.
OUTPUT RTN.
PRINT
*1
*2
*3
*4
*5
*4,*5
EXPLANATIONS
RTN ROUTINE (PROGRAM)
TB TABLE
CARD(INPUT)
PRINT(OUTPUT)
INFORMATION FLOW
PROGRAM CONTROL

Fig. 6 Program structure of MAFLOS

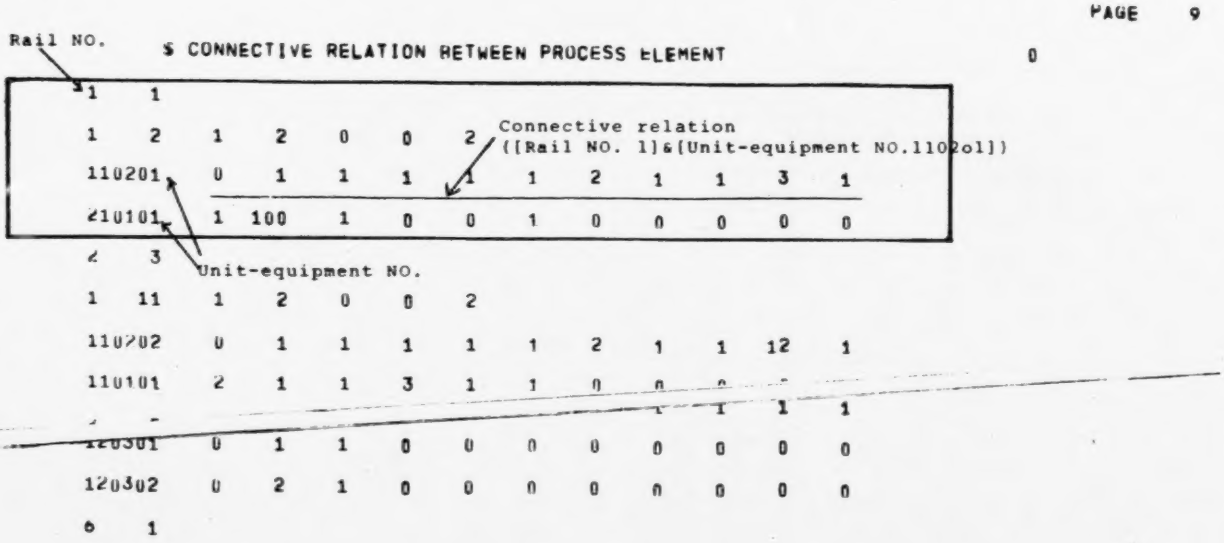
Table 3 Input data format for unit-equipment
(partial input data for the model shown in Fig. 1)



```
                                                                      PAGE    6
          $ ATTRIBUTE OF PROCESS ELEMENT                                   0
           * STORAGE                                                       6
    110101   3   1   1
    110201   1   3   1
    110202   1  20   1
    110203   1   3   1
    110204   1   3   1
    110205   1   3   1
           * PRODUCTION MACHINE                                            4
    1   3   1
    1   5
    2   5
    3   5
    120101   0   1   1   4   0   0   0   1   1   2   0   0  10   0
    0   0   0   0   0   0   0   0   0   0
    2   3   1
    1   5
    2   5
    3   5
    120201
```

Storage NO.
Storage capacity (X,Y,Z)
Work center NO.
Production machine NO.
Production machine ability

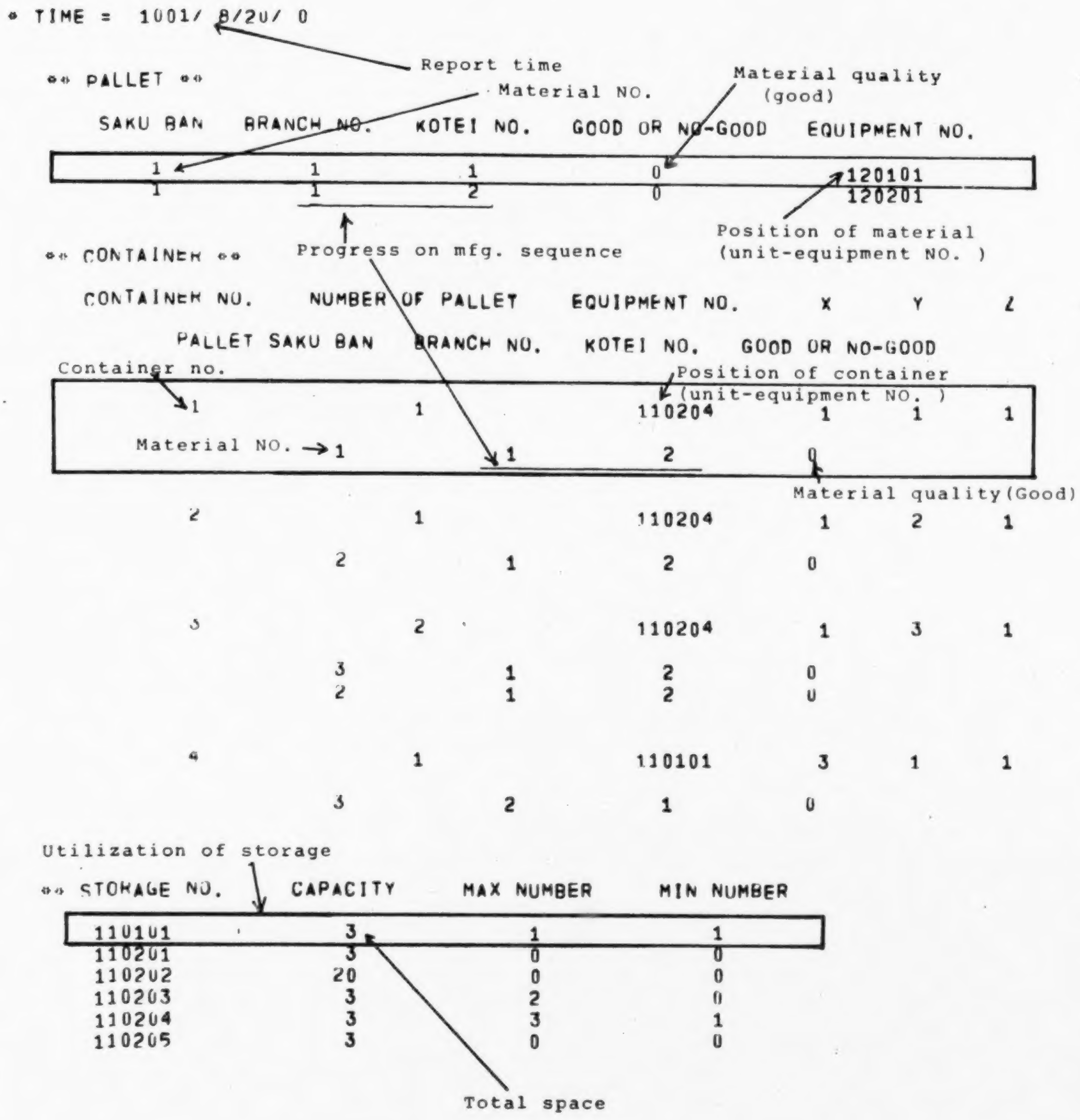
Table 4 Input data format for connective relation
(partial input data for the model shown in Fig. 1)



```
                                                                      PAGE    9
          $ CONNECTIVE RELATION BETWEEN PROCESS ELEMENT                    0
    1   1
    1   2   1   2   0   0   2
    110201   0   1   1   1   1   1   2   1   1   3   1
    210101   1 100   1   0   0   1   0   0   0   0   0
    2   3
    1  11   1   2   0   0   2
    110202   0   1   1   1   1   1   2   1   1  12   1
    110101   2   1   1   3   1   1   0   0   0   [illegible]
    [illegible]                                  1   1   1   1
    120301   0   1   1   0   0   0   0   0   0   0   0
    120302   0   2   1   0   0   0   0   0   0   0   0
    6   1
```

Rail NO.
Connective relation
([Rail NO. 1]&[Unit-equipment NO.110201])
Unit-equipment NO.

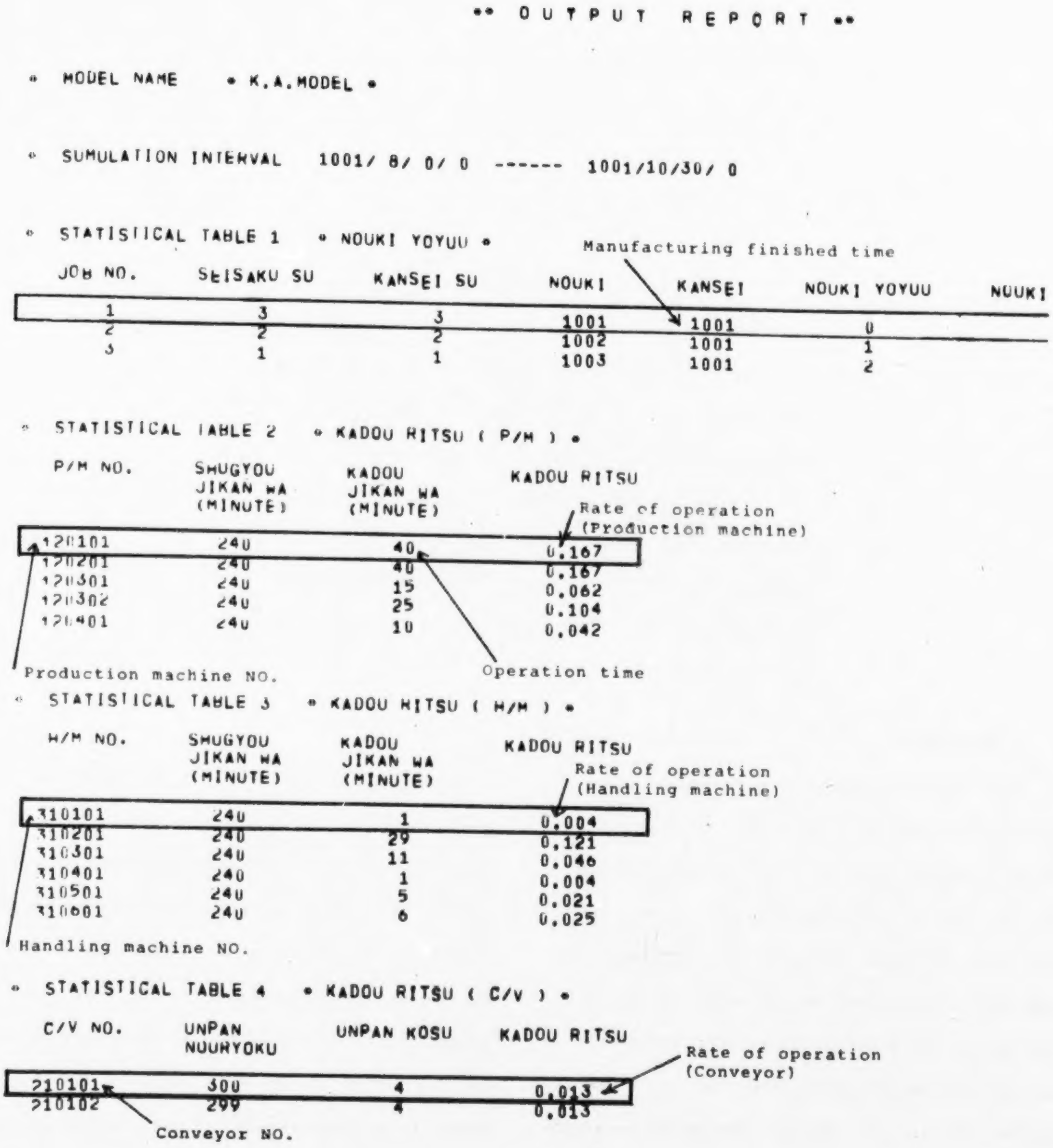
Table 5   Interim simulation report for the model shown in Fig. 1



```
* TIME =  1001/ 8/20/ 0

  ** PALLET **

    SAKU BAN   BRANCH NO.   KOTEI NO.   GOOD OR NO-GOOD   EQUIPMENT NO.

       1           1            1              0              120101
       1           1            2              0              120201

  ** CONTAINER **

    CONTAINER NO.    NUMBER OF PALLET    EQUIPMENT NO.     X      Y      Z

            PALLET SAKU BAN    BRANCH NO.    KOTEI NO.   GOOD OR NO-GOOD

         1                 1                 110204        1      1      1

                  1              1             2         0

         2                 1                 110204        1      2      1

                  2              1             2         0

         3                 2                 110204        1      3      1

                  3              1             2         0
                  2              1             2         0

         4                 1                 110101        3      1      1

                  3              2             1         0

  ** STORAGE NO.     CAPACITY     MAX NUMBER     MIN NUMBER

     110101              3              1              1
     110201              3              0              0
     110202             20              0              0
     110203              3              2              0
     110204              3              3              1
     110205              3              0              0
```

Report time
Material NO.
Material quality
(good)
Progress on mfg. sequence
Position of material
(unit-equipment NO.)
Container no.
Position of container
(unit-equipment NO.)
Material NO.
Material quality(Good)
Utilization of storage
Total space

Table 6 Final simulation report for model shown in Fig. 1



```
                                   ** O U T P U T   R E P O R T **

    * MODEL NAME     * K.A.MODEL *


    * SUMULATION INTERVAL   1001/ 8/ 0/ 0  ------  1001/10/30/ 0


    * STATISTICAL TABLE 1    * NOUKI YOYUU *
      JOB NO.       SEISAKU SU      KANSEI SU       NOUKI      KANSEI    NOUKI YOYUU      NOUKI
         1              3               3           1001        1001          0
         2              2               2           1002        1001          1
         3              1               1           1003        1001          2


    * STATISTICAL TABLE 2    * KADOU RITSU ( P/M ) *
      P/M NO.       SHUGYOU         KADOU           KADOU RITSU
                    JIKAN WA        JIKAN WA
                    (MINUTE)        (MINUTE)
      120101          240              40             0.167
      120201          240              40             0.167
      120301          240              15             0.062
      120302          240              25             0.104
      120401          240              10             0.042

    * STATISTICAL TABLE 3    * KADOU RITSU ( H/M ) *
      H/M NO.       SHUGYOU         KADOU           KADOU RITSU
                    JIKAN WA        JIKAN WA
                    (MINUTE)        (MINUTE)
      310101          240               1             0.004
      310201          240              29             0.121
      310301          240              11             0.046
      310401          240               1             0.004
      310501          240               5             0.021
      310601          240               6             0.025

    * STATISTICAL TABLE 4    * KADOU RITSU ( C/V ) *
      C/V NO.       UNPAN           UNPAN KOSU      KADOU RITSU
                    NUURYOKU
      210101          300               4             0.013
      210102          299               4             0.013
```

Manufacturing finished time

Rate of operation (Production machine)

Production machine NO.

Operation time

Rate of operation (Handling machine)

Handling machine NO.

Rate of operation (Conveyor)

Conveyor NO.

A DESCRIPTION OF AN AAW MODEL AND ITS CLASSROOM USES.

Alvin F. Andrus

Naval Postgraduate School

Monterey, California

Abstract

A probabilistic event store computer simulation of the interactions between surface-to-air missile systems and aircraft in a non-jamming environment and over flat terrain is presented. The purpose of the model is to test the general disposition of the missile areas and the associated missile system reaction times against an aircraft attack. The model is used as text material in a simulation course. Several model applications are included.

1. INTRODUCTION

The model presented in this paper is an event store computer simulation of the interactions between surface-to-air missile systems and aircraft in a non-jamming environment and over flat terrain. The model is programmed in FORTRAN. The purpose of the model is to test the general disposition of missile areas and the associated missile system reaction times against an aircraft attack. The model is a probabilistic monte carlo simulation. That is, the success or failure of a probabilistic event is determined in the model by comparing the numerical value assigned to the probability of success or failure to a program generated random number. The model was constructed as a classroom aid to be used in a graduate course on system simulation as applied to military conflict situations. The motivation behind the construction was to provide a model that would be complex enough to be interesting for the student to use and at the same time simple enough to illustrate the programming techniques of computer simulation model building.

2. PLAYING AREA

The playing area for the model is a pie slice portion of a circle. The center and radius of the circle and the central angle defining the pie slice are inputs. The numerical restrictions within the computer program are such that the central angle and radius must be less than 180 degrees and 1000 miles respectively.

3. OFFENSE

The offense consists of as many as twenty aircraft. These aircraft fly through the playing area in an attempt to penetrate a set of missile defenses. The entry points into the playing area for the aircraft are generated uniformly over the arc of the circle defined by the playing area. The flight path for each aircraft after it enters the playing area is to fly straight toward the center, (GX,GY). The spacing time between aircraft and the speeds and altitudes of aircraft are generated uniformly between their respective minimum and maximum values. These minimum and maximum values are inputs to the model.

The aircraft in the model play a passive role and serve only as the set of stimuli needed to cause the missile systems to act. These aircraft do not defend themselves against missile attack nor do they attack the missile areas.

4. DEFENSE

The defense consists of as many as three missile areas with their associated missile systems. These missile areas need not be located within the playing area; however, since only the results of interactions occurring within the playing area are considered in the model, the sphere of influence of the missile area must include some portion of the playing area in order for the missile areas to exert any effect on the simulation results.

Associated with each missile area are the parameters needed to describe its missile system. The values of these parameters are inputs to the model, and the parameters are:

(1) Search radar maximum range.

(2) Missile maximum range.

(3) Missile average speed.

(4) The number of tracking radars.

(5) The number of missile launchers.

(6) Maximum and minimum time required to reload a launcher.

(7) Maximum and minimum time required to assess a target after missile intercept.

(9) Missile single-salvo kill probability.

The significant time delays inherent to the missile systems included in the model are:

(1) Reload time: The amount of time required to reload a missile launcher.

(2) Acquisition time: The amount of time required, once an aircraft is observed on the search radar, to transfer the aircraft as a target to an available tracking radar.

(3) Assessment time: The amount of time the tracking radar must remain trained

on the target after missile intercept in order for the result of the intercept to be observed.

In the model all of these times are assumed to be uniformly distributed between their maximum and minimum values, which are inputs to the model.

5. ASSUMPTIONS

It is an assumption of the model that all aircraft are observed by all missile areas subject to the aircraft radar horizon and the missile area search radar maximum range. It is also the case that in order to fire a missile, or salvo, at an aircraft:

(1) The aircraft must be observed at the time of fire.

(2) A missile launcher must be loaded.

(3) A tracking radar must be free in order to be used for full course missile guidance.

(4) The intercept point must be within the missile maximum range circle.

(5) The aircraft must not be past the point of closest approach to the missile area at the time of fire.

The firing doctrine for a missile system is shoot-look-shoot at all available aircraft. That is, when a missile area has launched a salvo against a target no new salvos against that target will be launched from that missile area until that salvo has intercepted the target and the results of the intercept have been assessed. The aircraft are selected as targets, within the missile launcher and tracking radar numerical restrictions, on a first-come first-served basis. The model does not include altitude or minimum range restrictions on the missile.

An illustration of the playing area with a typical missile area and aircraft flight path is included as Figure 1.

6. GAME DOCTRINE

With the input parameter values assigned the model considers the interactions that occur in the playing area between the missile systems and aircraft. For the given set of defensive and offensive parameters the required number of aircraft will enter the playing area at points, times, speeds and altitudes generated by the computer program. This set of aircraft will then proceed directly toward the center, (GX,GY), passing through the missile defenses.

One complete pass through the computer simulation with one set of aircraft is referred to as a replication. To generate data for statistical purposes, at the completion of a replication the computer program will generate a new set of aircraft and using the same set of input values will produce another replication. The desired number of replications is an input value and must be less than twenty-one. An entire set of replications for a given number of aircraft is referred to as a run. For each run the model output consists of any of the following forms of output:

(1) Battle History: An event history of each replication containing the generated events of the battle in the

order in which the events occur and are generated.

(2) Standard: A compilation of each replication containing all aircraft initial conditions and the number of salvos fired by each missile area at each aircraft and the identification of the missile area responsible for killing each aircraft.

(3) Summary: A summary of information, by totals with respect to replication, for each run including the sample mean, variance and standard deviation of all totals presented.

The computer program will make as many runs as desired with an increased number of aircraft for each run. The number of aircraft in the first run, the increment for the number of aircraft in each new run, and the number of runs are input values. Each new run is considered by the model to be an extension of the previous run, that is, if run three contained seven aircraft and run four is to contain nine aircraft, then for all replications in run four the first seven aircraft will have entry points, altitudes, speeds and times identical to those replications in run three, etc. The random numbers used in the replications of a run in order to determine the outcome of probabilistic events are used again in the replications of a new run. In this manner it is hoped that any changes in the results between runs can be attributed to the increase in the number of aircraft rather than to the deviations of the sets of random numbers used. The model contains two missile firing procedures. These procedures are referred to as uncoordinated and coordinated and the procedure used is determined by the user as an input to the model. The uncoordinated missile firing procedure allows all missile areas in the simulation to fire missiles at all aircraft that can possibly be fired upon while the coordinated missile firing procedure allows a missile area to fire missiles at an aircraft only if no other missile area is currently engaging that aircraft. When the user elects to employ both procedures, they are not intermixed in the simulation but are run separately and the same sets of aircraft and sequences of random numbers are used in the corresponding replications and runs of the simulation so that differences in the results can be attributed to the procedure used.

7. EVENTS

As mentioned earlier the model is an event store computer simulation, i.e., all actions that are to occur in the simulation are dynamically generated by the computer program as a result of previous simulation actions and are listed chronologically in an Event Store List. Each of the actions included in the simulation assumes the form of a computer program subroutine, called an event, and the information pertaining to the action on the Event Store List is the information needed to execute the proper subroutine. There are only four major actions included in the model as events and these events are:

(1) Fire Missile Salvo.

(2) Missile Intercept.

(3) Reload Missile Launcher.

(4) Free the Tracking Radar from an Intercepted Target.

Each of the computer program subroutines representing these events uses as input parameters the following information:

(1) Time event is to occur.

(2) Identification of Event.

(3) Identification of Aircraft.

(4) Identification of Missile Area.

The dynamic process of simulating one air battle from start to finish forms the executive routine for the computer simulation. This executive routine consists of two program subroutines referred to as SNE and TNE. SNE, Store Next Event, is the subroutine that takes the generated information pertaining to an interaction and properly places this information on the Event Store List. TNE, Take Next Event, is the subroutine that, at the completion of any of the four events, interrogates the information on the Event Store List and transfers control of the computer program to the proper subroutine.

General flow charts describing the logic included in each event of the simulation plus the interrelationship of events are included as Figure 2 through Figure 6.

8. MODEL RESULTS

In this section a typical application of the model is presented. Basic to this discussion are the set of model inputs contained in Table 1. The position of the missile sites is illustrated in Figure 7. The measure of effectiveness used in this presentation is missile system effectiveness defined as the percent of aircraft killed averaged over the replications. Using this basic input as a starting scenerio we shall use the model to investigate trade offs in the values of the missile system parameters in an effort to maintain missile system effectiveness at a minimum value of .95.

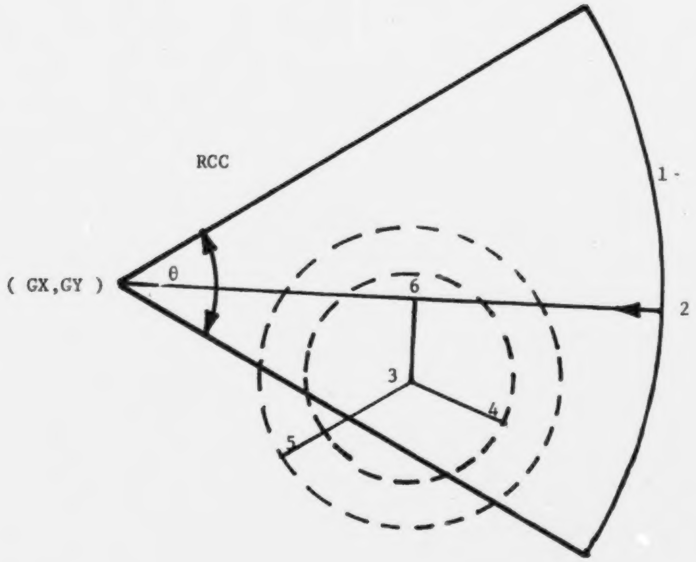
8.1 Missile Kill Probability: In order to determine an effective minimum acceptable missile kill probability for the missile system the missile kill probability was varied from 35 to 95 percent while all other parameters were held constant. The results of the model, i.e. the percent of aircraft killed as a function of missile kill probability for four raid sizes, are displayed in Figure 8. As expected, the percent of aircraft killed increases with increasing missile kill probability.

In Figure 9 is the graph of the percent of aircraft killed as a function of raid size for the missile kill probabilities of 35, 65 and 95 percent. From the graph it can be seen that for each of these missile kill probabilities the saturation raid size for the missile system appears to be between 10 and 15 aircraft, i.e. the percentage of aircraft killed seems to begin decreasing in this range indicating the missile system begins to lose effectiveness for raid sizes larger than 10. It can also be seen that there isn't much difference between the coordinated and uncoordinated firing modes. This is due to the position of the missile sites and the range of the missile in the scenario, i.e. these constraints are such that very few aircraft are simultaneously considered as targets by more than one missile site. It should be noted that for the 65 percent missile kill probability that missile system effectiveness is not at the desired level of 95 percent. Maintaining the missile kill probability at 65 percent, we shall now look at other parameters of the system to determine their effect on missile system effectiveness.

8.2 Missile Speed: The missile average speed was then varied from 600 to 1300 miles per hour. The effect on missile system effectiveness for the four raid sizes is graphed in Figure 10. The results indicate, again as expected, that the percent of aircraft killed increases as missile speed increases but is still below 95 for the raid size of 20. Figure 11 contains the graph of missile system effectiveness as a function of raid size for the selected missile speeds 600, 900 and 1300 miles per hour.

8.3 Aircraft Speed: Employing a missile speed of 1300 miles per hour and a missile kill probability of 65 percent the sensitivity of the system was tested against aircraft speed. The model was run varying aircraft speed from 350 to 1050 miles per hour. The results are graphed in Figure 12. Figure 13 contains the graph of missile system effectiveness as a function of raid size for the selected aircraft speeds 350, 750 and 1050 miles per hour. It can be seen from these graphs that missile system effectiveness decreases as aircraft speed increases and that for the aircraft speed of 750 miles per hour the missile system effectiveness has decreased below 90 percent for all raid sizes tested.

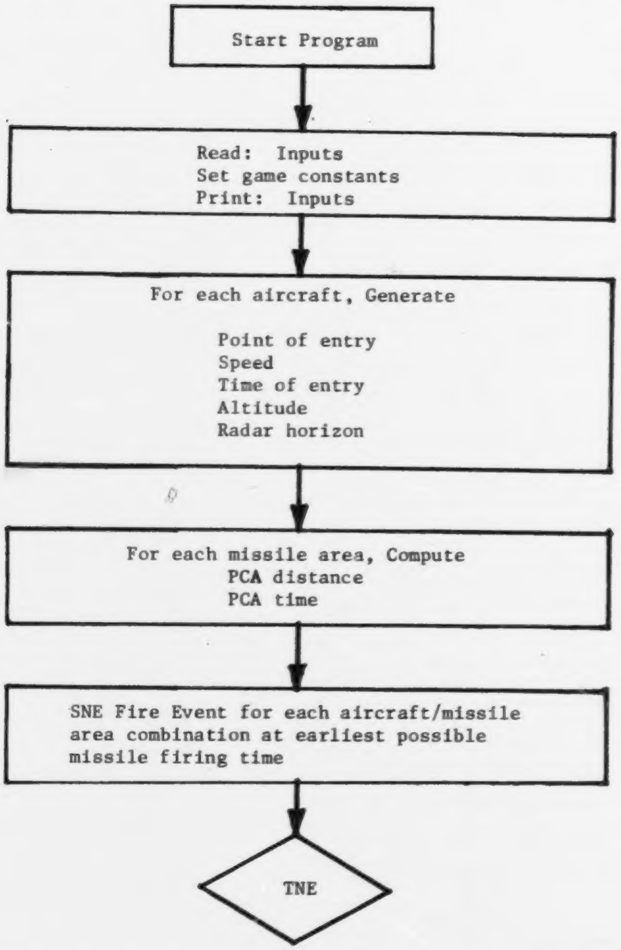
8.4 Tracking Radars and Launchers: Using the same scenario as that used above for the sensitivity of the system with respect to aircraft speed, the basic missile system was changed from one launcher and two tracking radars to two launchers and four tracking radars at each site. The results are graphed in Figure 14. This increase in missile system capability provides an increase across the board in missile system effectiveness. Figure 15 contains the graph of missile system effectiveness as a function of raid size for the selected aircraft speeds of 350, 750 and 1050 miles per hour. When comparing these results to those contained in Figure 11 it should be noted that the "doubling" of missile system capability does not in fact double missile system effectiveness. At an aircraft speed of 750 miles per hour for instance, the maximum increase in missile system effectiveness caused by the increase in missile system capability is 45 percent. The overall maximum increase in missile system effectiveness is 73 percent and occurs at an aircraft speed of 1050 miles per hour with a raid size of 20.



1:	Entry arc for aircraft
2:	Typical aircraft entry point and flight path
3:	Location of missile site
4:	Missile maximum range circle
5:	Search radar maximum range circle
6:	Point of closest approach
GX,GY:	Center of circle defining playing area
RCC:	Playing area radius
θ:	Central angle

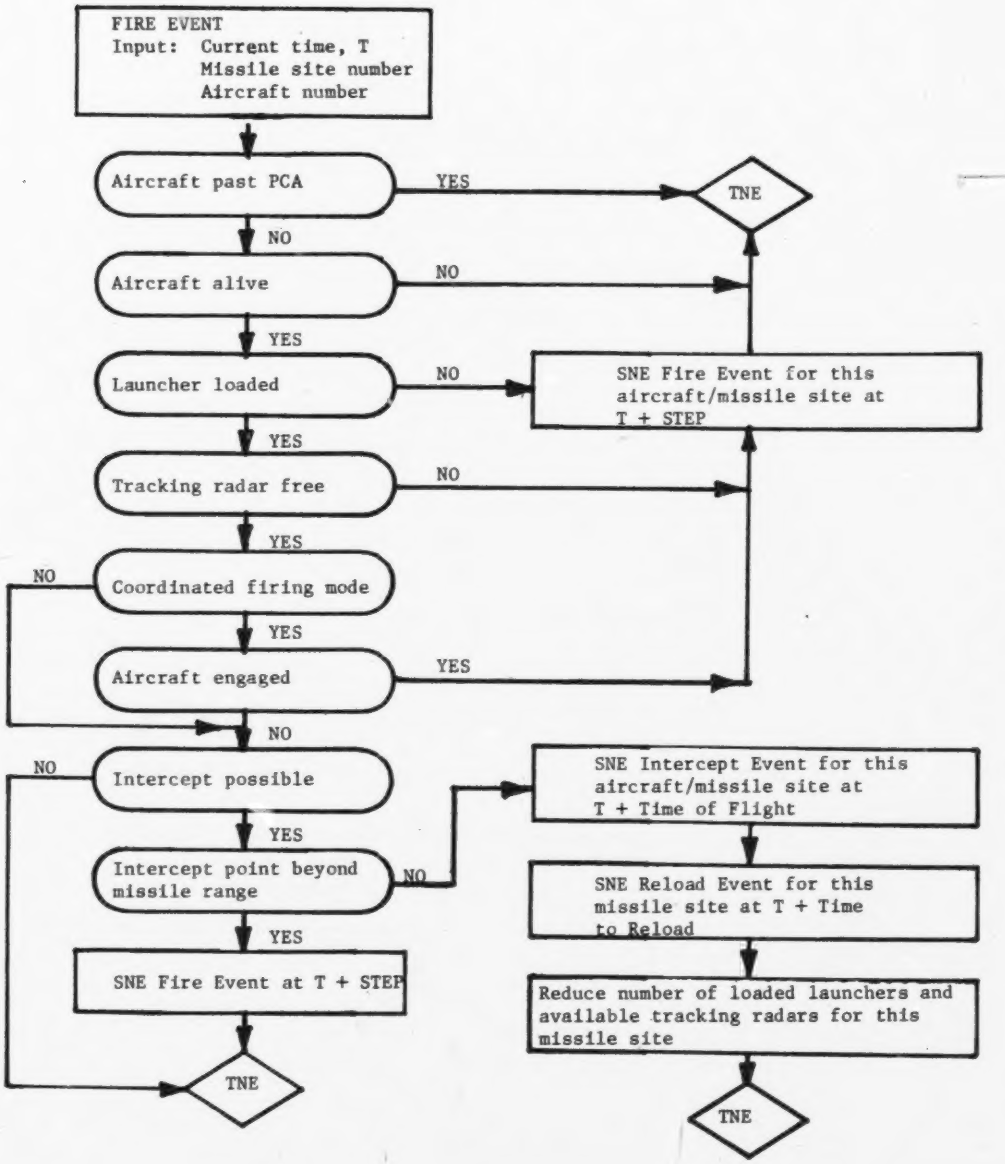
Playing Area Illustrating Missile Site and Aircraft Flight Path

Figure 1


Start Program
Read: Inputs
Set game constants
Print: Inputs
For each aircraft, Generate
Point of entry
Speed
Time of entry
Altitude
Radar horizon
For each missile area, Compute
PCA distance
PCA time
SNE Fire Event for each aircraft/missile area combination at earliest possible missile firing time
TNE

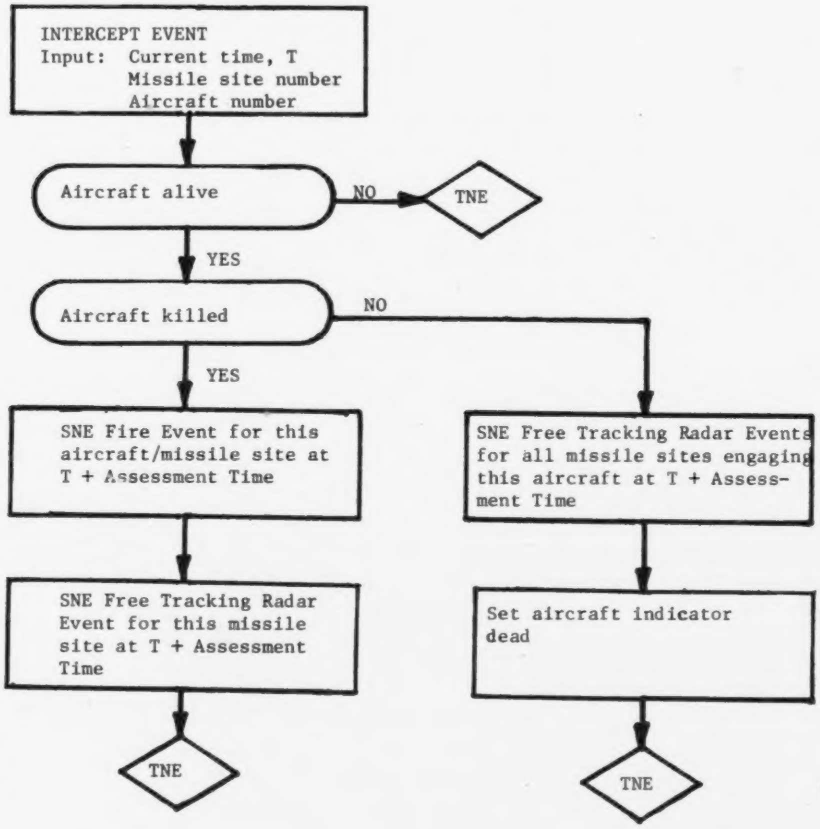
Simulation Logic for Model Initialization

Figure 2


FIRE EVENT
Input: Current time, T
Missile site number
Aircraft number
Aircraft past PCA
YES
TNE
NO
Aircraft alive
NO
YES
Launcher loaded
NO
SNE Fire Event for this aircraft/missile site at T + STEP
YES
Tracking radar free
NO
YES
NO
Coordinated firing mode
YES
Aircraft engaged
YES
NO
NO
Intercept possible
SNE Intercept Event for this aircraft/missile site at T + Time of Flight
YES
Intercept point beyond missile range
NO
SNE Reload Event for this missile site at T + Time to Reload
YES
SNE Fire Event at T + STEP
Reduce number of loaded launchers and available tracking radars for this missile site
TNE
TNE

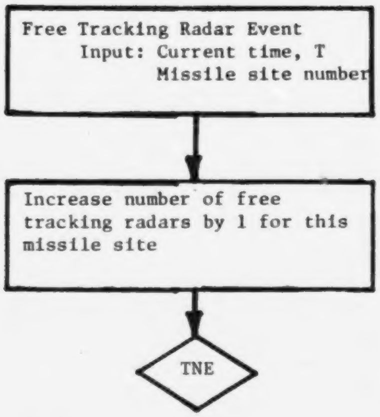
Fire Event Logic

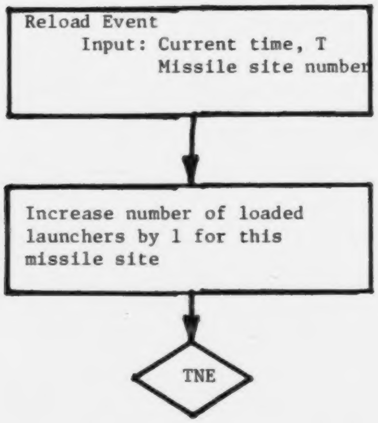
Figure 3


INTERCEPT EVENT
Input: Current time, T
Missile site number
Aircraft number
Aircraft alive
NO
TNE
YES
Aircraft killed
NO
YES
SNE Fire Event for this aircraft/missile site at T + Assessment Time
SNE Free Tracking Radar Events for all missile sites engaging this aircraft at T + Assessment Time
SNE Free Tracking Radar Event for this missile site at T + Assessment Time
Set aircraft indicator dead
TNE
TNE

Intercept Event Logic

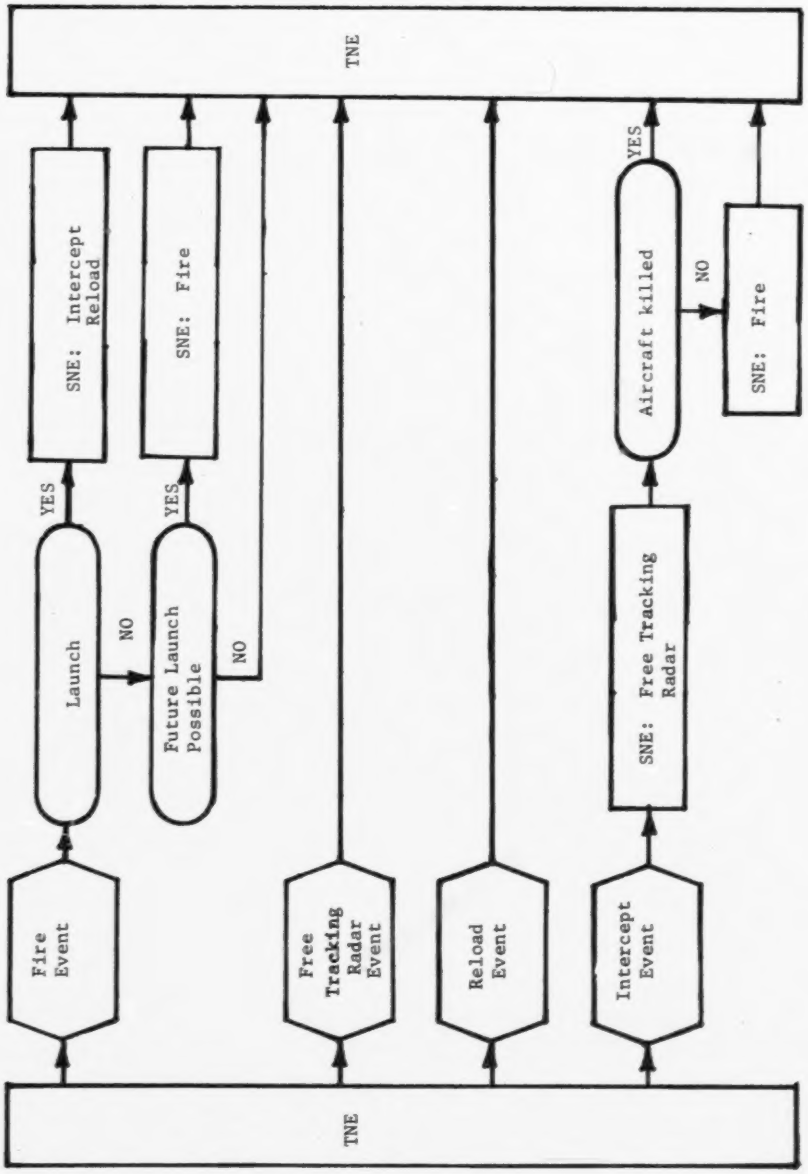
Figure 4


Free Tracking Radar Event
Input: Current time, T
Missile site numbe
Increase number of free tracking radars by 1 for this missile site
TNE


Reload Event
Input: Current time, T
Missile site numbe
Increase number of loaded launchers by 1 for this missile site
TNE

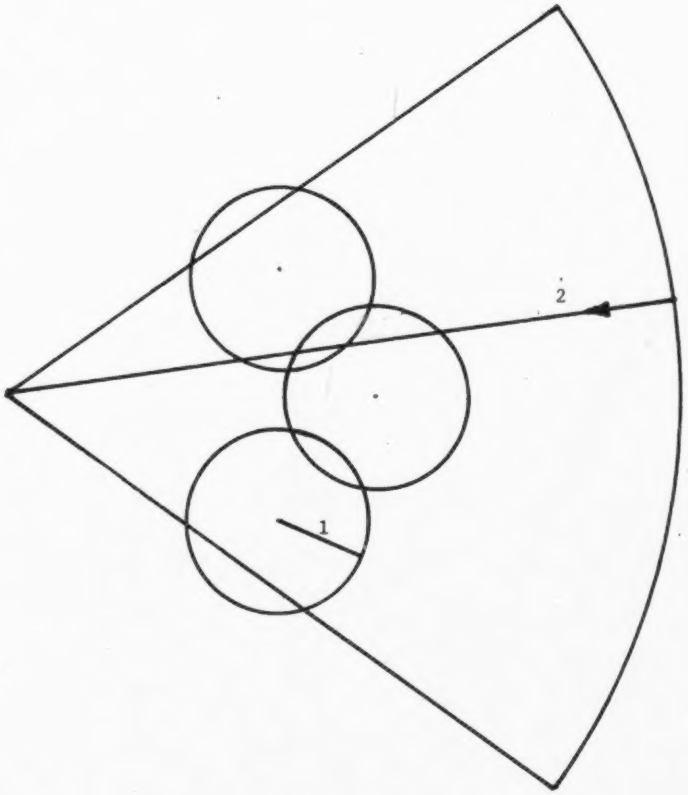
Free Tracking Radar Event Logic
Reload Event Logic

Figure 5


TNE
Fire Event
Launch
YES
SNE: Intercept Reload
NO
Future Launch Possible
YES
SNE: Fire
NO
Free Tracking Radar Event
Reload Event
Intercept Event
SNE: Free Tracking Radar
Aircraft killed
YES
NO
SNE: Fire
TNE

Interrelationship of Events

Figure 6



1: Missile maximum range, 50 miles
2: Typical aircraft flight path

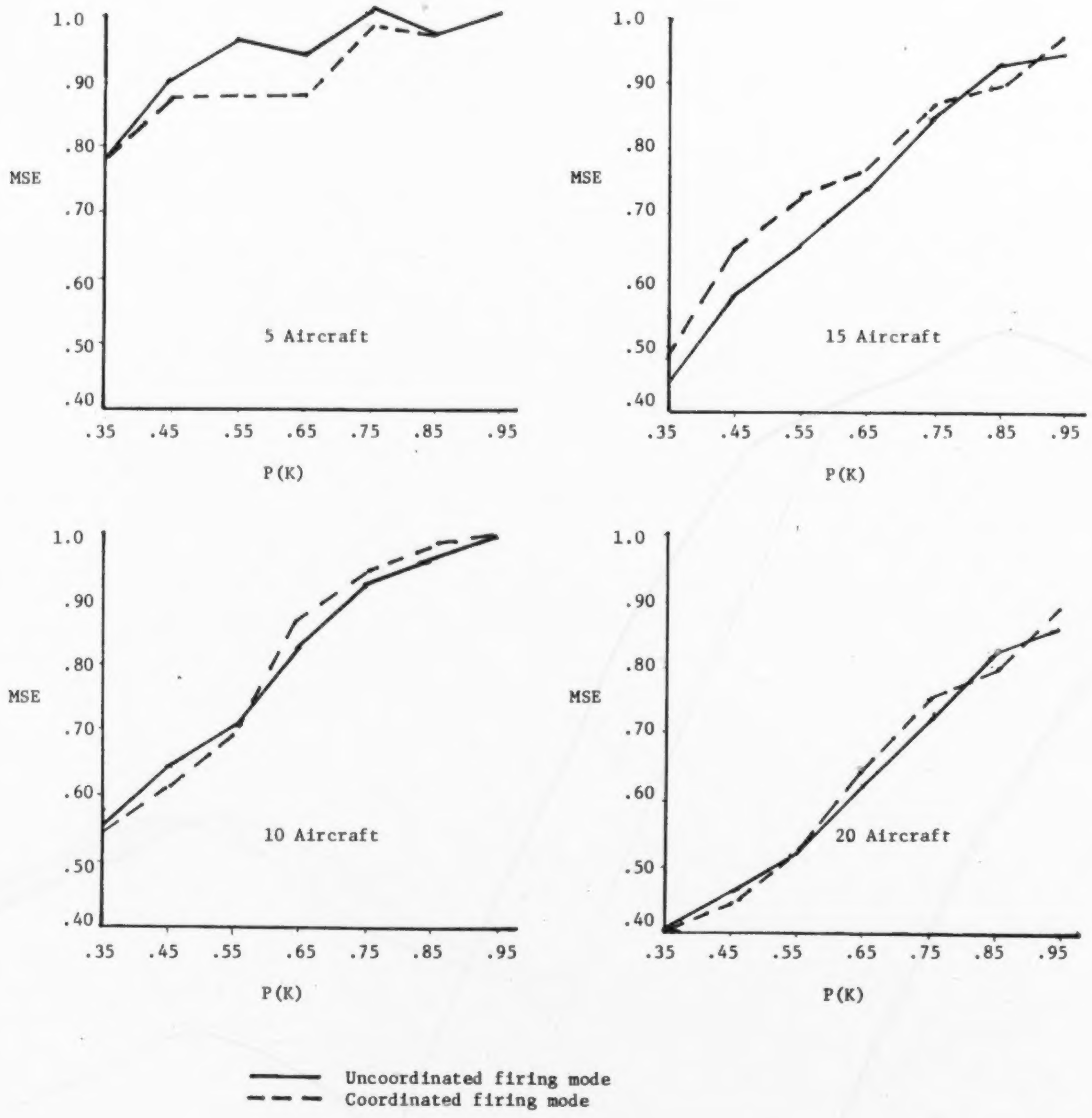
Disposition of Missile Sites for Application Scenario

Figure 7

```
INPUT DATA

   GAME PARAMETERS

         XCOORDINATE OF CENTER (GX) =      0.0   MILES
         YCOORDINATE OF CENTER (GY) =   500.00   MILES
         RADIUS OF CIRCLE     (RCC) =   500.00   MILES
         CENTRAL PLAYING ANGLE (CPA) =   90.00   DEGREES

 MISSILE AREA INFORMATION

  MISSILE AREA NR        1          2          3

                      130.00     160.00     130.00      MILES       AX     = XCOORDINATE
                      570.00     500.00     430.00      MILES       AY     = YCOORDINATE
                         1          1          1                    MAL    = NR OF LAUNCHERS
                         2          2          2                    MTRF   = NR OF T RADARS
                      200.00     200.00     200.00      MILES       RMR    = SEARCH RADAR MAX RANGE
                       50.00      50.00      50.00      MILES       RMAX   = MISSILE MAX RANGE
                      600.00     600.00     600.00      MI/HR       AVS    = MISSILE AVG SPEED
                        0.17       0.17       0.17      MINS        ATM    = ACQUISITION TIME MINIMUM
                        0.75       0.75       0.75      MINS        ATX    = ACQUISITION TIME MAXIMUM
                        0.17       0.17       0.17      MINS        ASM    = ASSESSMENT TIME MINIMUM
                        0.33       0.33       0.33      MINS        ASX    = ASSESSMENT TIME MAXIMUM
                        0.17       0.17       0.17      MINS        RTM    = RELOAD TIME MINIMUM
                        0.50       0.50       0.50      MINS        RTX    = RELOAD TIME MAXIMUM
                       65.00      65.00      65.00      PER.        PK     = KILL PROBABILITY PER SALVO

 AIRCRAFT INFORMATION

       MAXIMUM SPEED (SSX)= 350.0   MILES/HR
       MINIMUM SPEED (SSM)= 350.0   MILES/HR
       MAXIMUM ALTITUDE (AMAX)=    2000.0  FEET
       MINIMUM ALTITUDE (AMIN)=    2000.0  FEET
       MAXIMUM SEPARATION TIME (HTX)=   0.02  MINS.
       MINIMUM SEPARATION TIME (HTM)=   0.0   MINS.
```

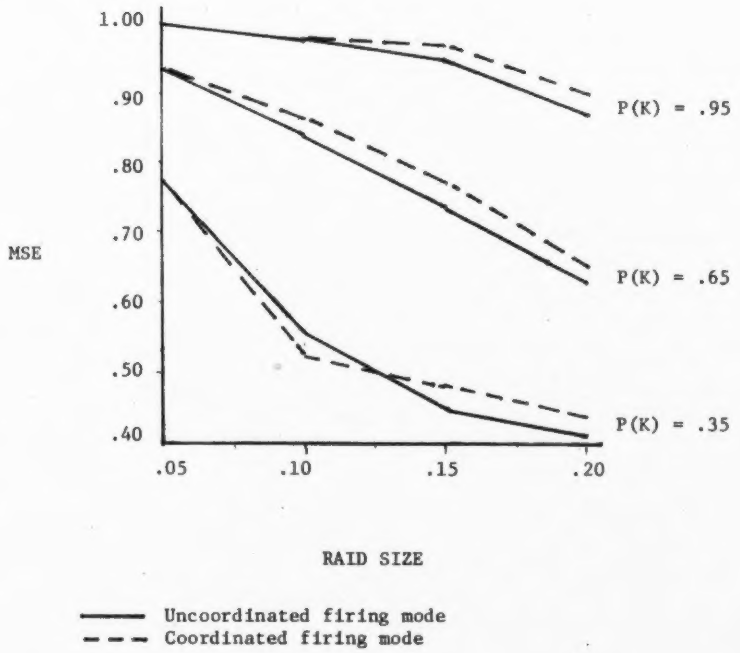
Input Information for Application Scenario

Table 1



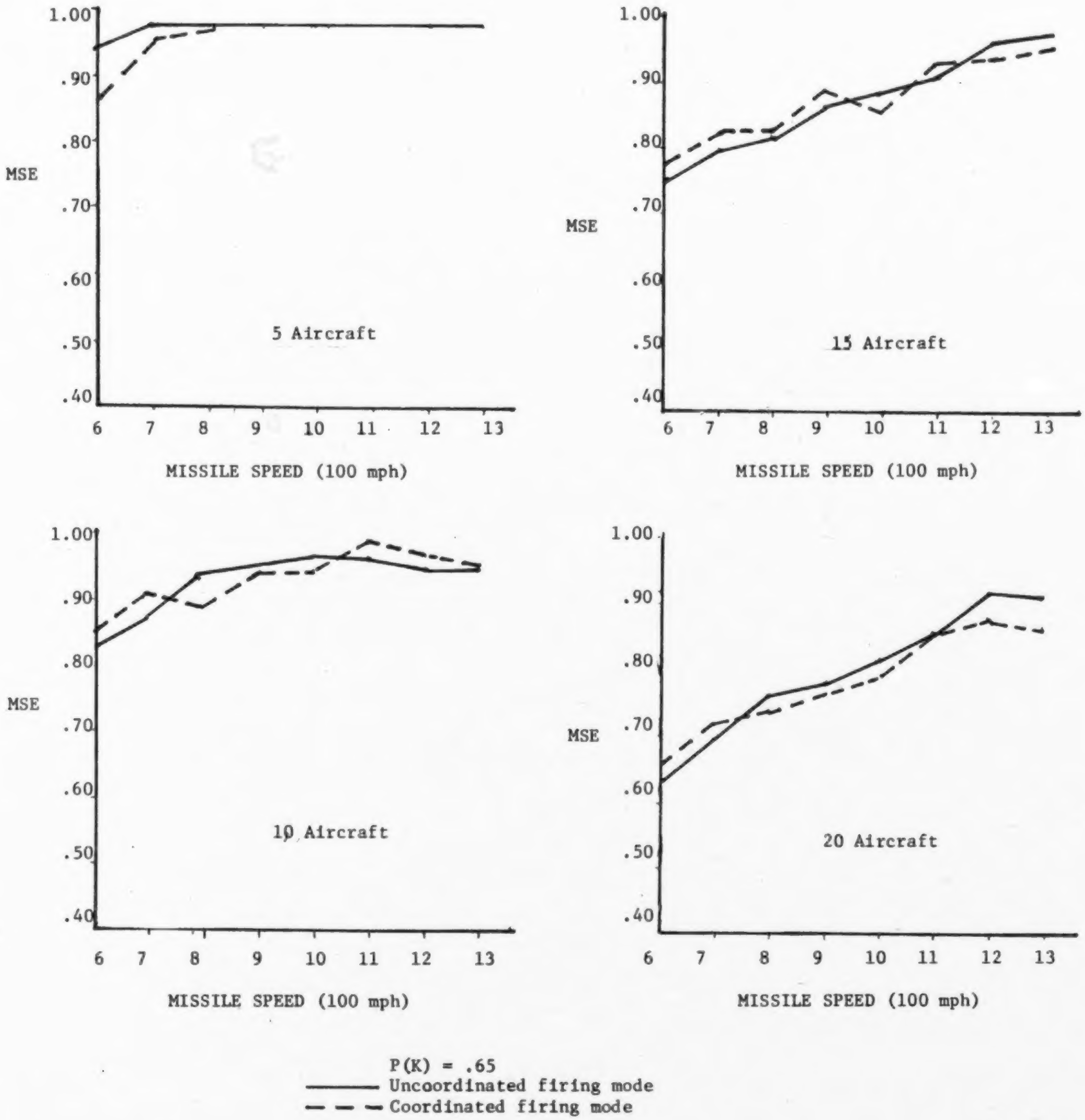
Missile System Effectiveness (MSE) vs Missile Probability of Kill (P(K))

Figure 8



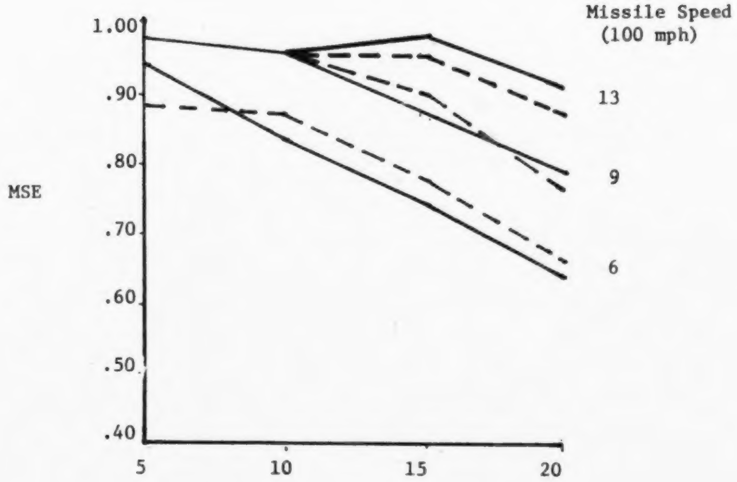
Missile System Effectiveness (MSE) vs Raid Size

Figure 9



Missile System Effectiveness (MSE) vs Missile Speed

Figure 10



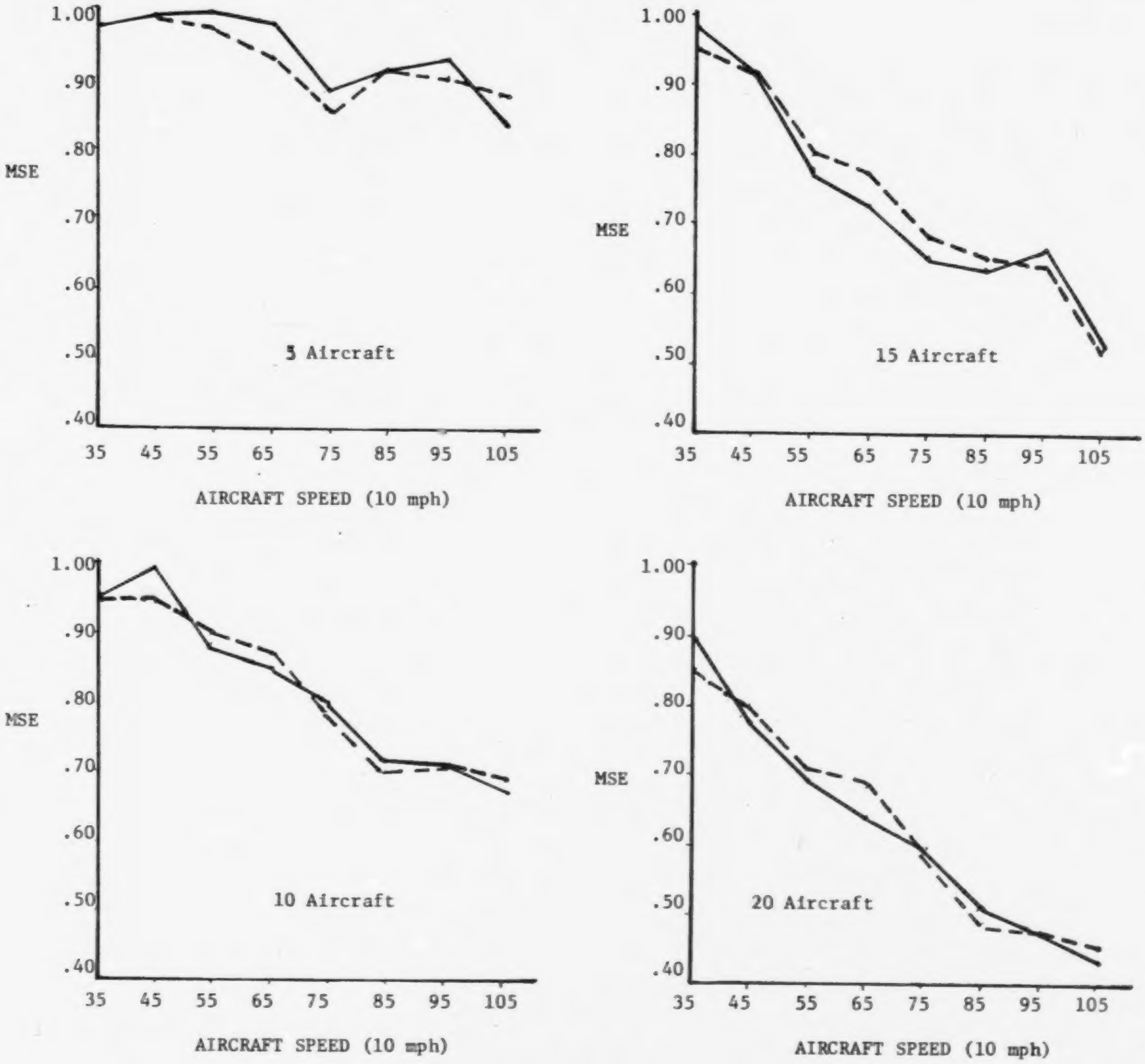
RAID SIZE

P(K) = .65

─── Uncoordinated firing mode
– – – Coordinated firing mode

Missile System Effectiveness (MSE) vs Raid Size

Figure 11



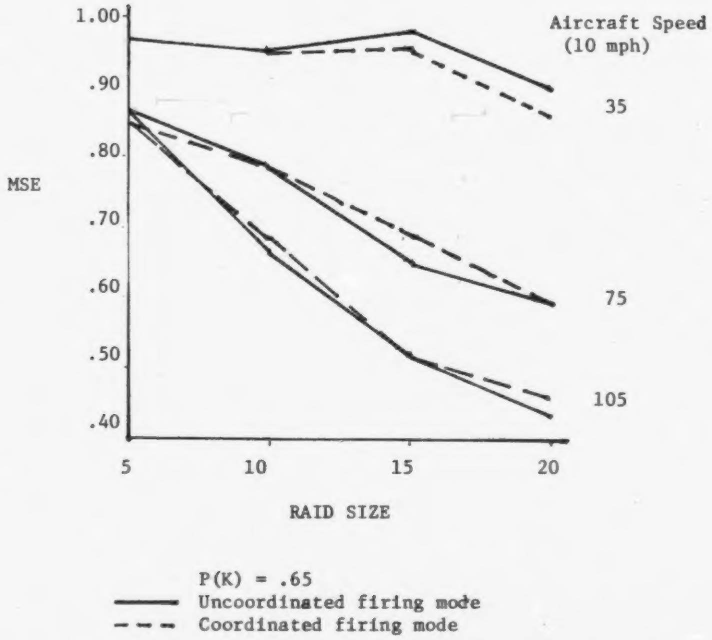
P(K) = .65

Missile Speed = 1300 mph

—— Uncoordinated firing mode
– – – Coordinated firing mode

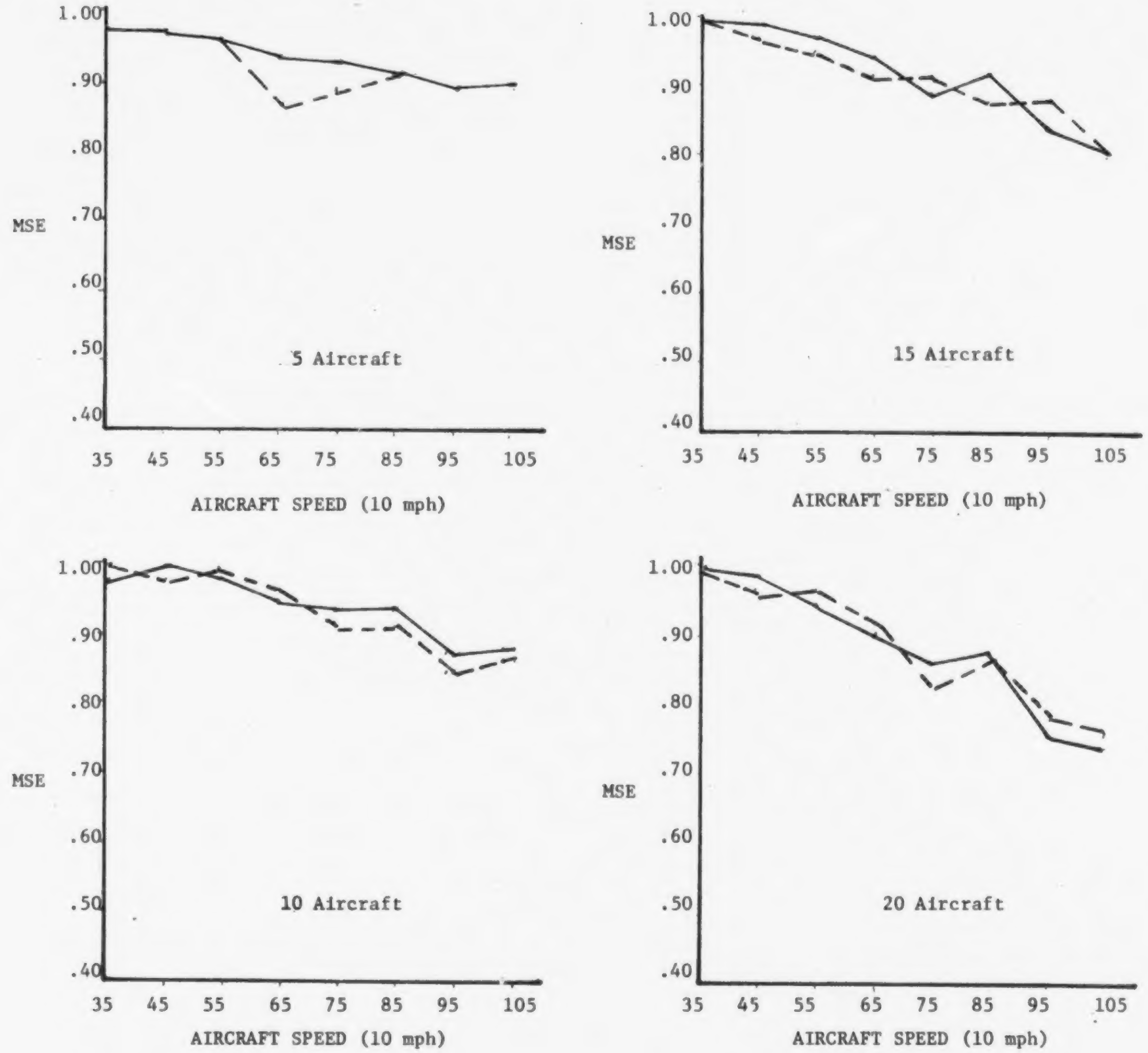
Missile System Effectiveness (MSE) vs Aircraft Speed

Figure 12



Missile System Effectiveness (MSE) vs Raid Size

Figure 13



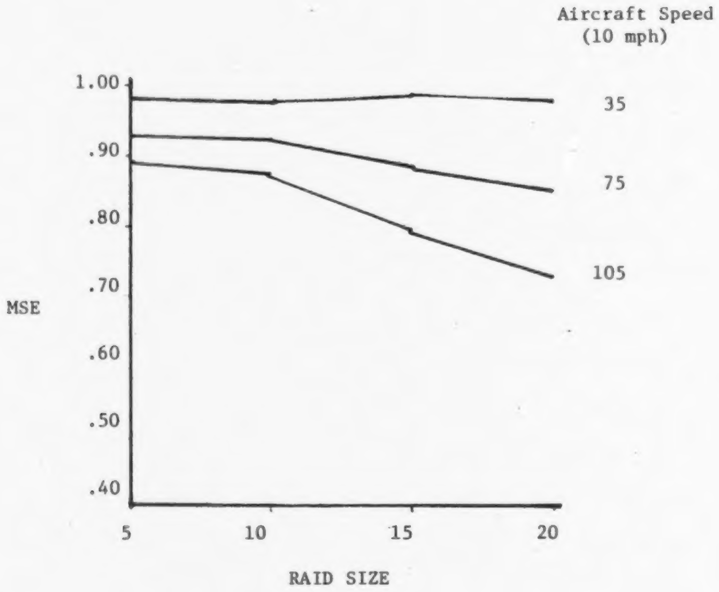
P(K) = .65
Missile Speed = 1300 mph

—— Uncoordinated firing mode
– – – Coordinated firing mode

Number of Tracking Radars per missile site = 4
Number of Launchers per site = 2

Missile System Effectiveness (MSE) vs Aircraft Speed

Figure 14



P(K) = .65
Missile Speed = 1300 mph

Number of Tracking Radars per missile site = 4
Number of Launchers per site = 2

Missile System Effectiveness (MSE) vs Raid Size

Figure 15

9. CONCLUSION

This paper has attempted to describe in minimum detail a missile system simulation and some typical applications. It has been assumed that the complexity of the surface-to-air missile anti-air warfare situation is such that answers to the questions posed in the applications of Section 8 are not readily available by convenient analytical methods. If this is true then a model of this type can serve a useful purpose. The model has been used in several classes as an aid to solving several anti-air warfare problems. In the course in system simulation in which the model is used the student adapts the model to a problem of his own selection, creates the inputs, uses the model to generate data and then performs an appropriate analysis of the data. The simplicity of the model's structure has influenced the thinking of several students in the development of models for Master's Thesis in Operations Research at the Naval Postgraduate School.

BIBLIOGRAPHY

1. Andrus, A. F. A computer simulation to evaluate surface-to-air missile systems in a clear environment. Naval Postgraduate School. Technical Report/Research Paper No. 67. 1966.

SIMULATION IN THE DESIGN OF AUTOMATED AIR TRAFFIC CONTROL FUNCTIONS

Paul D. Flanagan, Judith B. Currier, Kenneth E. Willis
METIS Corporation

Abstract

This paper describes the design and use of a simulator of some of the newly automated safety separation functions for terminal air traffic control (ATC). The program was used not only for analysis and design of these functions but also as a testbed for the logic actually implemented in the Knoxville, Tennessee terminal. Imbedded in the program is an emulator of the Goodyear Aerospace Corporation STARAN IV Associative Processor used at Knoxville. The three major ATC functions simulated are: 1) advanced mid-air conflict prediction and evaluation, 2) conflict resolution maneuver generation, and 3) automated voice advisory message generation and scheduling.

INTRODUCTION

In early 1971, the Federal Aviation Administration (FAA) began a program to provide expanded automated air traffic control (ATC) functions at the FAA test site in Knoxville, Tennessee. At that time, there was an ongoing program (the ARTS III program) to provide automated radar tracking and alphanumeric display functions at the 63 largest terminals in the U.S. The Knoxville experiment was designed to extend these automated functions to provide safety separation functions, namely, mid-air conflict prediction, resolution maneuver generation, and automated vocal traffic advisories.

Another important aspect of the Knoxville experiment was the evaluation of the STARAN associative processor (AP) built by Goodyear Aerospace Corporation. This new type of computer was to be considered as an addition to the ATC system to provide large amounts of computational power for various specialized ATC functions. The STARAN would act in conjunction with a UNIVAC 1230 processor to provide the data processing required for the Knoxville terminal area.

As participants in this program, the authors designed and used a simulation of the ATC system to be used at Knoxville. The simulator was used for analysis and design of the safety separation software required for the experiment. In addition, the simulator functioned as a testbed for the logic actually implemented in Knoxville.

The major benefits derived from designing and testing the software on the simulator were low cost of implementation and ability to allow parallel software development. Many functions were being programmed for Knoxville, a unique system with only limited time available for program test. By designing and testing the safety separation logic on the simulator, other functions could use more time on the Knoxville system without impeding logic development. In addition, the simulation program was located in the programming facility in the Washington, D.C. metropolitan area. Thus, costs of simulation development were offset by decreased travel costs and reduction in time required on the Knoxville computers.

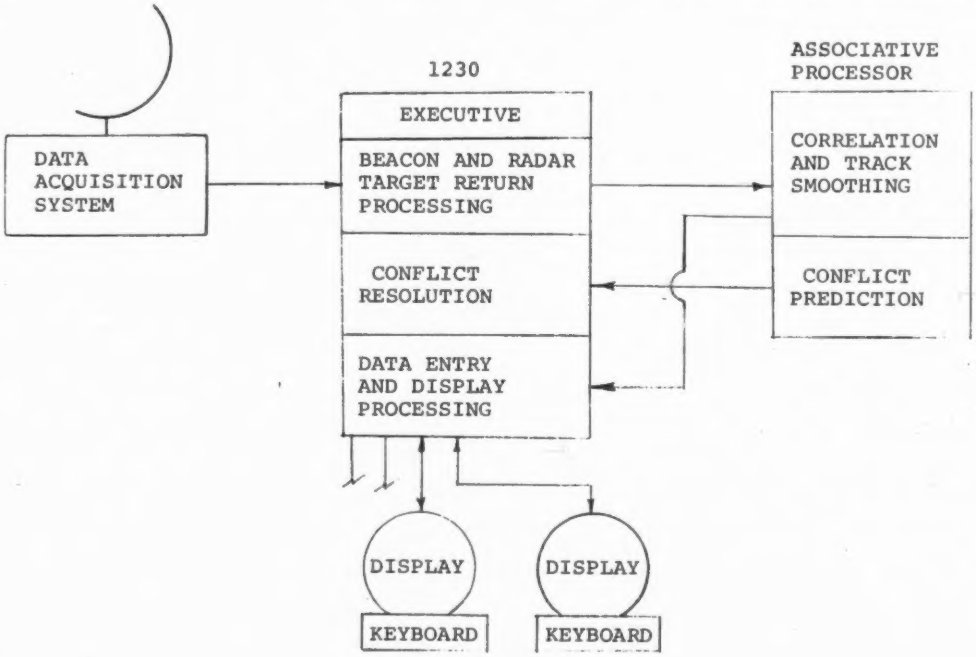
GENERAL DESIGN

The simulator was a model of the software functions used in the Knoxville experiment. No hardware was simulated explicitly, although hardware characteristics which impacted on the software functions were included in the model. Figure 1 displays the data processing functions of the original Knoxville experiment.

The simulator was written in FORTRAN for the CDC 6600 computer. The functions of the Executive, Beacon and Radar Target Return Processing, and Data Entry and Display Processing were modeled functionally. That is, the input and output of these functions were defined by the Knoxville system and used in the simulator. The logic of these simulated functions, however, was not identical to that used in Knoxville. The Conflict Resolution function was modeled logically. The simulator was used as a test bed for this function. The software of the Associative

Processor was modeled by using an emulator of the AP in the simulation. The same instructions used in the AP at Knoxville were input to the emulation portion of the simulator to provide these functions. In addition to the functions shown on Figure 1, a test track generator was added to the simulator so that the output of the data acquisition system could be simulated.

As the project progressed, various changes were made in the required functions. Conflict resolution was changed so that traffic advisories could be automatically transmitted on the ATC voice channel. Conflict prediction was examined as a serial processor function rather than as a parallel processor function. Thus, various modules not shown on Figure 2 were added to the basic simulation design.



1230
ASSOCIATIVE PROCESSOR
EXECUTIVE
DATA ACQUISITION SYSTEM
BEACON AND RADAR TARGET RETURN PROCESSING
CORRELATION AND TRACK SMOOTHING
CONFLICT RESOLUTION
CONFLICT PREDICTION
DATA ENTRY AND DISPLAY PROCESSING
DISPLAY
KEYBOARD
DISPLAY
KEYBOARD

Figure 1

DATA PROCESSING FUNCTIONS - KNOXVILLE EXPERIMENT

Figure 2 shows the simulation structure. The box labeled APEX represents the associative processor emulator. This figure represents the initial simulation design. These modules were: serial computer tracking logic, serial computer conflict prediction, and automated voice advisory message generation.

The remaining sections of this paper

describe the design of the various simulation modules.

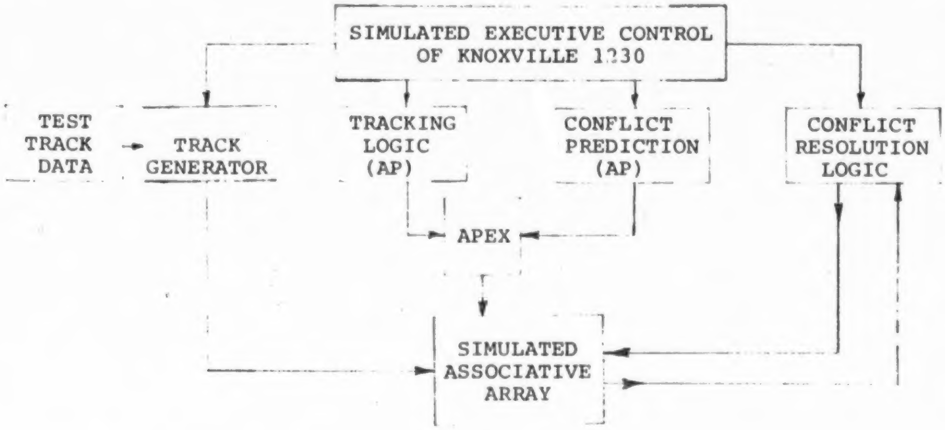

SIMULATED EXECUTIVE CONTROL OF KNOXVILLE 1230
TEST TRACK DATA
TRACK GENERATOR
TRACKING LOGIC (AP)
CONFLICT PREDICTION (AP)
CONFLICT RESOLUTION LOGIC
APEX
SIMULATED ASSOCIATIVE ARRAY

Figure 2

Simulation Structure

EXECUTIVE AND TEST TRACK GENERATOR

These two modules were the simplest modules in the simulator. The executive function merely called the other modules in the order they were called at Knoxville. The system worked on a radar scan time basis. Each scan represented four seconds of time, equal to one radar antenna rotation. The track smoothing logic was called eight times a scan (once every half second of simulated time) and all other functions were called once per scan.

The test track generator merely read in a description of the desired test track flight paths. At each scan, the position of each aircraft would be updated to a new position. This new position would be reported to the tracking function after a suitable amount of error was added to model the radar position errors.

ASSOCIATIVE PROCESSOR EMULATOR

This section describes the design of the module which functionally simulates the Goodyear Associative Processor, STARAN IV. Although the simulation was designed as a research and development tool, primarily for use in assisting in software development for the Knoxville experiment, the simulator is flexible in design, and is capable of accepting any algorithms written in the associative processor instruction set. It can easily be adapted to accept new or modified AP instructions.

The simulator system accepts as inputs a program written in the associative processor instruction set. The system then interprets and executes the AP instructions so as to provide outputs identical to those that the AP would provide. The simulator also can output the exact configuration of the associative array at any desired point in the AP program. Finally, it provides as output the exact time that would be consumed by the AP in executing the programmed instructions. (These timing calculations include the time required to page new instructions or data into the control memory of the AP.) The simulator is capable of maintaining statistics on the execution sequence in order to identify the time binds in the actual AP.

The Goodyear Associative Processor accepts programs written in an assembly language form. An assembler program executed on a XDS Sigma V computer translates such programs into the machine language instructions required by the AP. In order to make the simulator faster running, and be more useful, it was designed to accept as inputs programs written primarily in the higher order AP assembly language. The simulator maintains a simulated associative array in the core of the host computer. This array is manipulated by the AP instructions precisely as the AP manipulates its associative array.

The AP simulator is designed to operate in two parts. The first part is an assembler or encoder (GAPE - Goodyear Associative Processor Encoder). This program takes AP assembly language instructions as input and produces an interpretative code to be executed by the second part of the simulator. The second part of the simulator actually manipulates the simulated associative memory by executing the interpretative code produced by GAPE. The second part is named APEX for Associative Processor Executor. This division of the simulator is analagous to the standard assemble (or compile) and execute steps usually used in any higher level language. GAPE assembles the code into executable form and APEX loads the interpretative code and simulates the functions of the associative processor. Figure 3 displays the operational flow for AP algorithm execution.

The assembly step (GAPE) reads STARAN IV code directly from card input and translates the instructions into an "executable" form suitable for processing by the AP executer (APEX). The output is normally punched cards.

Additionally, the assembler flags errors in the STARAN IV input deck, such as illegal instructions, doubly defined symbols, and undefined symbols. The GAPE assembler is written in FORTRAN IV

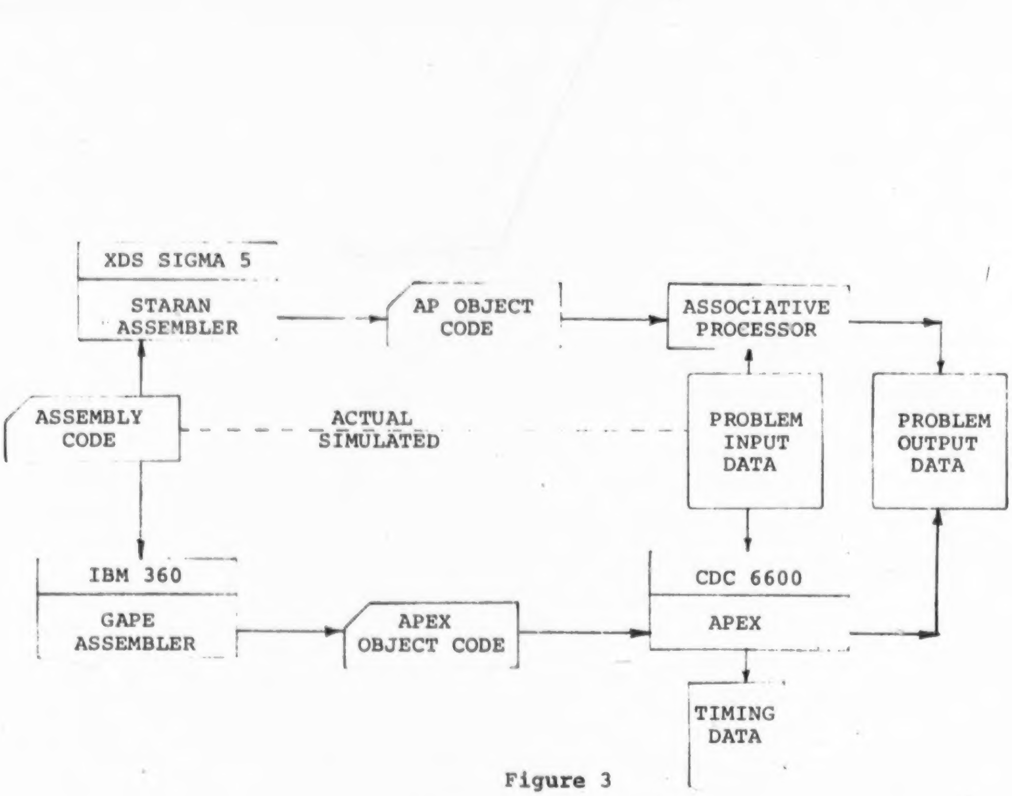

XDS SIGMA 5
STARAN ASSEMBLER
AP OBJECT CODE
ASSOCIATIVE PROCESSOR
ASSEMBLY CODE
ACTUAL
SIMULATED
PROBLEM INPUT DATA
PROBLEM OUTPUT DATA
IBM 360
GAPE ASSEMBLER
APEX OBJECT CODE
CDC 6600
APEX
TIMING DATA

Figure 3

Problem Solution Process

and BAL for the IBM 360.* It occupies approximately 80K bytes of core. Its device requirements are a card reader, a card punch, a printer, and a scratch file. The CPU time required is approximately .1 sec per AP assembly instruction on a 360/65.

The output deck of the assembler is a series of 30 bit words (punched in octal format). The general form of an operation to be performed by APEX is:

If T: then: A .op. B(I)$\rightarrow$S(J)
: otherwise: no operation

where T is a parameter to be tested to determine if the instruction is to be executed,

	A	is the first parameter used for the operation,
	.op.	is the binary operation to be performed,
	B	is the second parameter used for an operation,
	I	is the location of the "address" of B if indirect addressing is used,
	S	is the parameter field to be replaced by the results of the operation,
and	J	is the location of the "address" of S if indirect addressing is used.

---

*An alternate FORTRAN version is implemented on a CDC 6600.

The first word (30 bits) of each group of words transferred to APEX is called the command word. It contains basic instruction information. This information includes the instruction operation and suboperation codes (nine bits and four bits respectively), the number of parameter words following the command word (three bits), a parameter descriptor list (five bits), a data register index (five bits), an instruction test parameter (T) (one bit), an argument usage parameter (one bit), and the two most significant bits of the argument if present. If the argument usage bit is set, the command word is followed by a word containing the 30 least significant bits of the argument. Following the command word (or argument word if present) is a series of 15 bit half-words, packed two per word. These parameter half-words each contain a parameter identifier (e.g. A) (two bits), the address of the most significant bit of the parameter field (eight bits), and the number of bits in the parameter field minus one (five bits).

The second part of the emulator, APEX, is implemented in the FORTRAN IV language on the CDC 6600. The code is written to be compatible with the FORTRAN implemented on the UNIVAC 1230 computer.

APEX reads the instruction list provided by GAPE, reads the timing information, and simulates the logical and arithmetic functions of the STARAN IV processor. Figure 4 presents a diagram displaying the relationships between the subroutines comprising APEX. The design of APEX allows the user to input five different AP algorithms for selective execution by an external control program. This mode of operation closely simulates the mode of operation of the AP at Knoxville. As each algorithm is selected, APEX modifies the simulated associative memory according to the assembly instructions which created the algorithm.

There are two basic kind of instructions for the AP. The first kind uses inputs from words of associative memory or the response store and produces output in the associative memory or response store. These instructions must be accomplished sequentially in the simulator through each simulated word of associative memory. Subroutine DOIT provides for execution of these instructions. There are 15 operations or tests which can be performed by the AP on the bit fields of associative memory. These are: less than, less than or equal, equal, not equal, greater than, greater than or equal, logical not, logical and, logical or, logical exclusive or, absolute value, add, subtract, multiply, and divide.

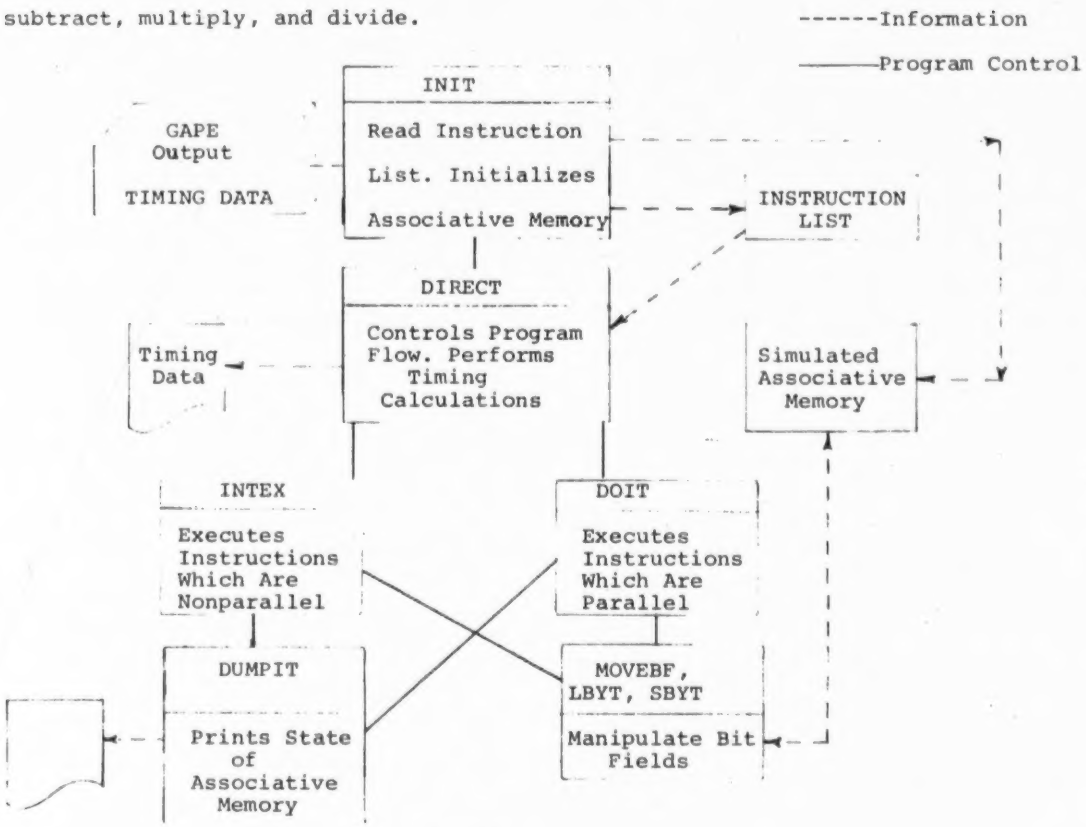

------Information
———Program Control
INIT
Read Instruction
List. Initializes
Associative Memory
GAPE
Output
TIMING DATA
INSTRUCTION
LIST
DIRECT
Controls Program
Flow. Performs
Timing
Calculations
Timing
Data
Simulated
Associative
Memory
INTEX
Executes
Instructions
Which Are
Nonparallel
DOIT
Executes
Instructions
Which Are
Parallel
MOVEBF,
LBYT, SBYT
Manipulate Bit
Fields
DUMPIT
Prints State
of
Associative
Memory

Figure 4

APEX Logic Relationships

Subroutine DOIT extracts the bit fields for each operand (A,B) from each word of associative memory; performs the requested operation, and stores the result in the specified field (S) in each associative memory word.

The other instruction type makes less use of associative memory. Instructions of this type manipulate the arithmetic register, output register, field pointers, or date registers. These instructions may also shift the response store values, control program flow, provide input/output, etc. This second type of instruction is executed by subroutine INTEX. This subroutine contains a small internal subprogram for each instruction of this type. As the instruction is encountered, the internal subprogram manipulates the AP registers in the appropriate fashion.

The use of DOIT and INTEX is controlled by subroutine DIRECT. This subroutine controls the selection of instructions and the interpreting of the command words and parameter identification words of the language produced by GAPE. Subroutine DIRECT also maintains the instruction execution frequency and timing information data.

Subroutine DUMPIT prints the state of the associative array on command. Subroutines MOVEBF, LBYT, and SBYT perform bit field manipulations in the simulated associative array.

CONFLICT PREDICTION

The prediction function is responsible for identifying (1) all pairs of aircraft that are in hazardous positions; and (2) all aircraft flying too close to the terrain. The basic design of this module was inspired by the implicit geometric filter concept described in the paper "Intermittent Positive Control" in the March 1970 issue of the Proceedings of the IEEE.

The airspace is divided into 1024 horizontal square cells, or "boxes", each 4 miles on one side. This grid covers a square area 128 miles to a side. Each aircraft is entered into a number of boxes according ot its location, speed, and direction of flight. The method of placement in cells is as follows:

1. Place the aircraft in the cell containing its present position. (This is the primary cell.)
2. If the aircraft is within 1/2 the minimum safe separation distance of a cell edge, place the aircraft in the adjacent cell. (This is a positional secondary cell.)
3. Place the aircraft in cells (called velocity secondary cells) according to three indices:
    a. speed class
    b. cell quadrant position
    c. heading

The speed class is an index to the aircraft's indicated ground speed. The cell quadrant position is the quadrant within the cell within which the aircraft lies. The heading is an index derived by dividing the compass into twelve equal sized (30 degree) sectors. This placement of the aircraft into adjacent cells to allow for flight during the warning time is performed by a table look-up on speed class, cell quadrant position, and heading. The table contains cell displacements to calculate the new cell indices from the basic cell index. The construction of the table was performed off-line to provide for fast real-time execution. The table considers the tracking errors.

After all aircraft are entered into the cells, the conflicts between these aircraft are selected. The box size and placement method are such that if an aircraft is the only cell occupant, there can be no conflict. If more than one aircraft occupies a cell, further tests are required to determine the hazard. (These tests are described later).

Terrain avoidance is performed during the cell placement process. Each cell has a minimum altitude for safe flight. Only aircraft with altitude information can be checked for terrain avoidance. Before an aircraft with altitude information is placed into a cell, its reported altitude is checked against the acceptable minimum altitude. If the aircraft is too low, the appropriate information is output.

There are three further filters to be performed before an aircraft pair is output as a conflict. First, the software determines if this aircraft pair was selected as a conflict in the previous 60 seconds. If so, no further processing is done, as this pair has already been processed by display or resolution. The second test is an altitude filter. If both aircraft have altitude information and have reported greater than 500 foot vertical separation, then the conflict is ignored. The final test is a coarse hazard filter. The coarse hazard filter adds the time dimension to the prediction process and determines if the aircraft can violate the safety standard.

Because the aircraft are tracked, the software can determine if a near miss is predicted. It is known that the heading data provided by the tracker is subject to error. Thus, for each speed class and bearing position, there is a maximum error in the velocity components in the two horizontal directions. The coarse hazard filter will add these maximal errors to the predicted velocities to obtain the highest relative closing velocity. Then, a simple miss distance calculation will determine if the aircraft could pass within the minimum safe miss distance.

CONFLICT RESOLUTION

Once a possible conflict has been isolated by the prediction function, it must be further evaluated to determine its relative collision potential, or risk. Ordering the possible conflicts by risk allows the automated system to respond consistently to the priorities of the users in presentation of warnings. The chosen measure of risk is the probability of violating a given miss distance within the warning time provided by the system. This probability is calculated from the geometric configuration of the two aircraft and the uncertainties inherent in their position and velocity data. The logic does not consider the conditional probability that an aircraft will turn from its current course, although future systems should utilize whatever "intent" information is available in the system.

Considering the aircraft's current position, velocity, and acceleration, it is possible to project an ensemble of possible paths which the aircraft could follow. The uncertainty associated with the choice of a path from this ensemble arises from two sources: variances in the current data, and uncertainty about the pilot's intent.

Uncertainty in velocity (in particular, heading) is a major source of spread in the path ensemble. This uncertainty is approximated by the normal distribution of straight paths symmetrically projected about the estimated heading of a non-turning track. Uncertainties in position are accommodated in the miss distance criteria.

Lack of knowledge of the pilot's intent is another source of uncertainty in defining the path ensemble. If an aircraft turns within the projected time period, then the assumption of straight flight can result in a hazard suddenly appearing with less than 60 seconds to possible impact. If all possible paths are included in the ensemble, however, the volume of airspace occupied by the ensemble grows, and data must be available to define the probability distributions of turning paths. The problem of turning aircraft is compounded by the fact that the tracking logic produces greater variances in current estimated heading, as well as time lags in heading prediction, when a turn is in progress. These considerations led to the assumption of a uniform distribution of headings in the direction of turn if a turn was determined to be in progress from the track data. The basis for the assumption was that if an aircraft were turning in the terminal environment, it was equally likely that it would continue turning, or stop turning at any point on the current trajectory (and proceed straight along a tangent to the turn curve). While this assumption is only a modest first approximation to a definition of the full path ensemble, it does reflect the broader distribution resulting from the turn in a realistic manner. Further development of the conflict resolution logic must examine the possibility of developing valid a priori probabilities of turn in the terminal airspace, perhaps as a function of aircraft position, wind patterns, or other variables. Future systems might incorporate such "intent" information from the other ATC functions in the data processor.

The risk probability is calculated by a numerical integration over the ensemble of possible paths of the two aircraft. The area encompassed by the potential paths is divided into a number of equal size segments, and a representative path selected for each segment. Each representative path has a probability associated with it. If the two aircraft paths result in a violation of the miss distance in the warning time, then the joint probability is summed into the risk probability. More explicitly, the risk probability becomes

$$\text{RISK} = \sum_{i} \sum_{j} p_i^a \, p_j^b \, \delta(D_{ca})$$

where p_i^a is the probability aircraft A traverses the i^{th} path in the ensemble and $\delta(D_{ca}) = 1$ if the distance of closest approach for paths i & j is less than a critical distance, and 0 otherwise. A more responsive risk criteria function currently under investigation would additionally weigh each contribution of a conflicting path pair according to the time remaining to violate the separation criteria. This criteria becomes

$$\text{RISK} = \sum_{i} \sum_{j} p_i^a \, p_j^a \, \delta(D_{ca}^{ij}) \times W(T_{ca}^{ij})$$

where $W(T_{ca}^{ij})$ is a weighting function (0-1) which is a function of T_{ca}^{ij}, the time of closest approach for paths i and j.

An aircraft configuration is considered hazardous if the risk probability is greater than a threshold value. According to the Knoxville experiment requirements, certain configurations required calculation of a maneuver which would eliminate the conflict. This maneuver was displayed to the controller for transmission to the pilot.

The aircraft are divided into two types, associated and unassociated. Associated aircraft are under direct positive control of an ATC controller (Instrument Flight Rules). Unassociated aircraft are not under positive control (Visual Flight Rules). For conflicts between one associated and one unassociated aircraft where both aircraft are reporting altitude information, a maneuver is calculated for the associated aircraft. For other types of conflicts, the controller is merely alerted to the existence of the conflict.

Note that when altitude information is available, the risk probability has been calculated on the basis of the three-dimensional position and velocity vectors. If a leveloff command will reduce the conflict probability below the threshold value, the recommended maneuver is "Level Off". However, if that probability is greater than the acceptable safety threshold, a turn maneuver is calculated. Since most of the aircraft in the system do not have altitude reporting transponders, positive climb and dive maneuvers are not generated. If a lateral maneuver is required, the associated aircraft is turned away from the unassociated aircraft until the distance of closest approach is greater than the allowable miss distance. The paths used to calculate this turn are selected from the ensemble of paths of the aircraft on the basis of shortest time to conflict.

The direction of turn may be determined by considering the two aircraft as a physical system and locating the positional centroid of this system at the time of the expected turn. The maneuvering aircraft is turned away from this centroid. Turning the aircraft towards the centroid may, in some cases, produce less severe maneuvers. However, in a system with uncertainties in aircraft position, velocity, and time of maneuver initiation, a turn toward the centroid often increases, not decreases, the risk. The time at which the relative location of the centroid, and therefore the advisable direction of turn, changes is different for any two aircraft paths. Thus the ensemble of paths around each aircraft generates a spectrum of these critical times. If, within the time span of interest, different paths have different advisable turn directions, the maneuver becomes ambiguous. This ambiguity can be discovered by inspecting the edges of the ensemble of paths. If a maneuver is ambiguous in this sense, the resolution algorithm indicates this on the display and does not recommend a resolution.

To decide how far the aircraft should turn, different degrees of turn, at the standard rate, are projected. The distance between the aircraft pair at the end of each trial turn is determined in order to insure that the aircraft do not violate the minimum miss distance while the maneuvering aircraft is turning. The path tangent to the turn circle is then checked against the unassociated aircraft's worst case heading vector (appropriately projected in time) and the distance of closest approach is calculated. If this distance, the closest that the two aircraft will ever get to each other if neither deviates from the given course, is greater than the minimum miss distance, then this is considered to be a feasible maneuver. The two new vectors then have their risk probability calculated. If this probability is less than the threshold, then the maneuver is accepted. If the probability is greater than the threshold, then the aircraft is turned further and checked again. This last step is repeated until a safe maneuver is found. No turn greater than 180° is considered.

Tables are maintained to determine if an aircraft gets into multiple conflicts. If so, consistent maneuver suggestions are calculated. If the multiple conflict occurs from the same side, then the larger of the bearing changes will be sent as the maneuver. If the conflict is from the opposite side, then a flag is set to indicate that no unambiguous maneuver was found.

ADVISORY MESSAGE GENERATION

The newest function added to the simulator is a simple module which prepares messages which could be transmitted to pilots involved in conflicts. A similar function was implemented and tested at Knoxville in 1972.

The messages consist of traffic advisories and service messages. Traffic advisories warn an aircraft of the location and heading of aircraft which could be in conflict. Service messages warn of restricted areas, terrain conflicts, loss of radar contact, etc.

The simulation module merely prepares a message in a form which could be sent to an automatic voice response unit for transmission to the pilot. In general, the time required to send an advisory will exceed the time for one antenna scan. The average message will take two or three scans to transmit. During peak periods, there will be messages waiting in a queue to be sent to the aircraft. The message generator must choose the order in which the messages are transmitted.

The basic structure of the system allows for a flexible priority scheme for message transmission. There are defined classes of messages which are assigned different priorities based on selections made by the controller. In addition, within these classes, priorities are established as appropriate. For example, the risk level provides a good priority measure within the traffic advisory class. Thus, the most hazardous conflicts will receive the first messages. Other messages, such as "all clear of traffic" are given priority based upon time in queue.

Session 9: Financial Models (General)
Chairman: Theodore Mock, University of California

This session reports upon several uses of simulation methodology in finance and accounting. Two papers consider the effect of variability, variability assumptions (e.g. normality), and uncertainty on measurement of investment returns. In addition two applications of simulation in financial planning and control will be discussed.

Papers
"Accounting Rate of Return vs. True Rate of Return:
Considering Variability and Uncertainty"
John V. Baumler, Ohio State University

"Variability Assumptions and Their Effect on Capital Investment Risk"
F. J. Brewerton, Louisiana Tech. University and
William B. Allen, United States Air Force

"A Computerized Interactive Financial Forecasting System"
Phillip M. Wolfe and Donald F. Deutsch, Motorola, Incorporated

"Multiple Capital Budgeting: A Simulation Model"
Thomas J. Hindelang and Andre Fourcans, Indiana University

Discussants
Eugene Comiskey, University of California
Mel Greenball, Ohio State University
Kenneth Siler, University of California
Harry Grossman, Security Pacific Bank

ACCOUNTING RATE OF RETURN
VS. TRUE RATE OF RETURN
CONSIDERING VARIABILITY AND UNCERTAINTY

J. V. Baumler
Associate Professor
Academic Faculty of Accounting
The Ohio State University

Abstract

An accounting rate of return and a defined true rate of return were assessed for a simulated firm composed of independent long-lived investment projects. The parameters of each individual investment project were determined by a Monte Carlo simulation technique. Differing degrees of environmental variability and uncertainty were represented by the simulation techniques used. Accounting rate of return, defined consistent with contemporary accounting practice, and a true rate of return, defined in economic terms, were contrasted. The efficacy of accounting rate of return as a surrogate for true rate of return was found to be a function of the degree of variability and uncertainty represented in the environment.

I wish to thank members of the Accounting Research Colloquium at The Ohio State University for their helpful comments and The Ohio State University Instructional and Research Computer Center for its support in making available free computer time.

The relationship between accounting rate of return (ARR) and true yield of a firm has been the subject of considerable research[1]. ARR has generally been defined consistent with accounting practice. True yield in such studies has been an economic concept. Previous researchers have contrasted accounting and economic measures in an almost endless variety of situations--and the accounting measures haven't always faired very well.

This paper starts with the conclusion that accounting measures of return are imprecise surrogates for the related economic concepts with which they are apparently to correspond. Such a conclusion, while disturbing to some, need not detract from the usefulness of accounting measures. It may be that period to period changes in accounting measures of return correspond closely with interperiod changes in true yield. The relationship between changes in ARR and changes in true yield is the subject of this paper.

Let us state the approach taken by means of an analogy. With a crude thermometer, we would not expect to accurately measure temperature, or even to record minor changes in temperature; but we would expect to be capable of identifying major changes in temperature. The precision of such a thermometer could be assessed by determining how violent temperature changes must be before they are capable of being recorded by the measuring instrument. In this paper, accounting and economic measures of return for a simulated firm, operating in a variable and uncertain environment, are defined and measured. Accounting measures can then be assessed in terms of the magnitude of actual underlying changes that are necessary to have a corresponding impact on the accounts.

In the next section of this paper, the model used to represent an enterprise, and the accounting and true yield measures are described. Particular emphasis is given to the means by which inter-year changes in the economic fortunes of the enterprise are induced. In the results section, the correlation between changes in the measured true yield and the measured ARR, as a function of inter-year variability, is presented. Finally, a macro-sensitivity analysis is presented to provide some indication of the generality of the results obtained.

The Model

As in most previous research, a firm will be envisioned as a collection of investment projects.[2] We will begin our description of the firm by describing an example project. This will be followed by a discussion of the generation and aggregation of projects.

An Example Project

It will help in describing projects to think of those aspects of a project which are known by management at the time it is undertaken, and those aspects which will be known at a later date. Consider the following example. At the time the investment project is undertaken, it is known that the investment outlay is $142,879 in then current dollars. The price index stands at 286. Management estimates the project will have a life of 8 years, after which time the salvage value will be $2,858. Straight line depreciation will be used for financial reporting purposes and sum-of-the-years-digits depreciation will be used for tax purposes. After the fact, the following information is known. The project actually lasted 10 years, at which time the salvage value was $8,200 but the price index stood at 410. The income tax rate was 50% throughout the project's life. The cash flow, expressed in real dollars on a before tax basis, generated from the investment each year and the associated price index is shown below.

Year	Real Dollar Before Tax Cash Flow	Price Index
1	$7549	297
2	6821	298
3	8749	298
4	6689	308
5	7842	322
6	6652	337
7	7010	356
8	8732	377
9	6497	389
10	8223	410

Real dollars refer to dollars with the purchasing power of an arbitrarily selected base year--the price index for that base year is 100. Given the ex ante and the ex post information, Tables 1, 2, and 3 can be prepared. Table 1 indicates what impact this project will have on accounting statements during its 10-year actual life and provides the very orthodox definition of ARR used in this study. Table 1 as well as Tables 2 & 3 are based on the following conventions. Investment outlays are made at the start of a year, other receipts and disbursements occur at the end of the year. Investments consist of depreciable assets only. The price-level index applies to the end of the year. Net receipts are either reinvested in other projects or distributed as dividends. With these conventions in mind, Table 2 can be seen as proof that the internal rate of return (IRR) for the example project is 4.56%. Table 3 presents earnings and asset values based on net present values using the IRR as a discount rate.[3] It presents measures of earnings and investment such that each year their ratio equals the IRR. Table 3 will be used below to define what will be called the true rate of return (TRR) for the firm.

True Rate of Return

From Table 1 and Table 3, the ARR and the IRR for this project can be compared. However, such a comparison is not very interesting. Rather, the comparison of the ARR and TRR for the firm is of interest. But if each project the firm undertakes has tables such as Table 1 and Table 3 prepared for it, the summation of appropriate table entries would provide the numerators and denominators of the two ratios of interest. This may be made clear by example. Assume the example project were undertaken in year 1961, and we are interested in the 1965 ARR and TRR for the firm as a whole. The numerator of the ARR would be the sum of appropriate entries from column (10) of tables like Table 1. The example project would contribute $3,893 to this sum. If each

TABLE 1

Calculation of Accounting Rate of Return for Example Project

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
			Reported Accounts and Ratios							
Years	Real dollar before tax cash flow	Price index (base = 100)	Current dollar before tax cash flow	Asset write-off using straight line depreciation	Asset balance-end of year	Average asset balance	Before tax profit	Provision for taxes	After tax profit	Accounting rate of return (10) ÷ (7)
1	7549	297	22444	17503	125376	134128	4942	2471	2471	1.84%
2	6821	298	20303	17503	107874	116625	2800	1400	1400	1.20%
3	8749	298	26069	17503	90371	99122	8566	4283	4283	4.32%
4	6689	308	20579	17503	72868	81620	3077	1538	1538	1.88%
5	7842	322	25288	17503	55366	64117	7785	3893	3893	6.07%
6	6652	337	22403	17503	37863	46614	4900	2450	2450	5.26%
7	7010	356	24926	17503	20360	29112	7423	3712	3712	12.75%
8	8732	377	32885	17503	2858	11609	15382	7691	7691	66.25%
9	6497	389	25270	0	2858	2858	25270	12635	12635	422.16%
10	10223	410	41896	2858	0	2858	39039	19519	19519	683.07%
				142879						

TABLE 2

Calculation of Internal Rate of Return for Example Project

Investment in Real Dollars = $142,879/2.86 = $50,000

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
			Accounts for Tax Purpose						
					Current Dollars				
Year	Real dollar before tax cash flow	Price index (base = 100)	Current dollar before tax cash flow	Asset write-off using sum-of-years digits depreciation method	Taxable income	Taxes paid	After tax cash flow	Real dollar after tax cash flow	Present value of (9) at 4.56%
1	7549	297	22444	31116	-8672	-4336	26780	9007	8614
2	6821	298	20303	27226	-6924	-3462	23765	7984	7303
3	8749	298	26069	23337	2732	1366	24703	8291	7252
4	6689	308	20579	19447	1132	566	20013	6505	5442
5	7842	322	25288	15558	9730	4865	20423	6333	5067
6	6652	337	22403	11668	10734	5367	17036	5058	3870
7	7010	356	24926	7779	17147	8573	16352	4599	3365
8	8732	377	32885	3889	28996	14498	18387	4882	3417
9	6497	389	25270	0	25270	12635	12635	3249	2174
10	10223	410	41896	2858	39039	19519	22377	5460	3495
									50000

TABLE 3

Calculation of True Rate of Return for Example Project

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Year	Real dollar after tax cash flow	True rate of return earnings .0456 x (6)	True rate of return capital recovery (2) - (3)	End of year Investment $(5)_{t-1}$ - (4)	Average investment $(5)_{t-1}$	True rate of return (3) ÷ (6)
1	9007	2281	6726	43274	50000	4.56%
2	7984	1974	6010	37264	43274	4.56%
3	8291	1700	6591	30673	37264	4.56%
4	6505	1399	5106	25568	30673	4.56%
5	6333	1166	5167	20401	25568	4.56%
6	5058	931	4128	16273	20401	4.56%
7	4599	742	3856	12417	16273	4.56%
8	4882	566	4316	8101	12417	4.56%
9	3249	370	2879	5222	8101	4.56%
10	5460	238	5222	0	5222	4.56%
			50000			

project had a Table 1, all the contributions to accounting earnings would be known. Similarly, the denominator of the ARR is obtained by summing appropriate entries from column (7). The example project contributes \$64,117. The numerator of the TRR is found by summing appropriate entries from column (3) of Table 3; and the denominator from column (6)[4]. The example project contributes \$1,166 and \$25,568 to the numerator and the denominator of the firm's 1965 TRR respectively. Thus, TRR as defined in this paper is the weighted average of the IRR's of the projects existing when the TRR is calculated. The weights used are asset values using the capital recovery method of depreciation[5].

To be more precise, let IRR_t be the internal rate of return for the project commenced in year t (this assumes 1 and only 1 project per year--an assumption relaxed later in this paper). Let $cash_{t,i}$ be the after tax, real dollar, cash flow in the i th year of the project commenced in year t. By convention, a positive value represents an inflow and a negative value (i=0) represents an outflow then, the true rate of return in year k is:

$$TRR_k = \frac{\sum_{t=1}^{k} IRR_t \sum_{i=0}^{k-t} - cash_{t,i}[1 + IRR_t]^{k-t-i}}{\sum_{t=1}^{k} \sum_{i=0}^{k-t} - cash_{t,i}[1 + IRR_t]^{k-t-i}}$$

TRR_k is totally expressable in terms of cash flows since IRR_t is the value for which

$$0 = \sum_i cash_{t,i}/[1 + IRR_t]^i$$

Another way of describing true rate of return and justifying that rather presumptuous label, is to take a brief look at the accounting problems of leasing companies. Two approaches have had widespread use in the field, the rental method and the financial method. The financial method, now in many cases required[6], is supported by analogy to economic concepts. In effect, TRR, as previously defined, is calculated using the financial method to account for non-lease investments. The financial method can only be used if all of the cash flows that will result from an investment are known. Such a requirement is approximately met in the leasing situation, and exactly met in a computer simulation. It could be met in the real world on a retrospective basis (indicating what the financial statements should have been) given adequate bookkeeping. Thus, the arguments for the validity of the TRR measure used in this paper are as strong as the arguments for the financial method of accounting for lessors. And a major problem from describing the TRR in terms of the capital recovery method of depreciation is countered. Some would argue that the capital recovery method of depreciation should use the firm's cost of capital as the discount rate; and not the IRR of the project with which the depreciable asset is associated. Such is not the case for the financial method of accounting for leases. The financial method uses the yield rate associated with each lease. Thus, the support for the financial method can be marshalled behind the TRR measure used in this study.

The Collection of Projects

Assume for the moment the existence of the simulated firm. Each year a certain after tax cash flow is generated (by convention, at the end of the year). The amount of cash to be invested (again by convention, at the start of the next year) is a random variable between 50% and 150% of the cash flow generated. If less than 100% is reinvested, it is assumed that the rest is distributed as dividends. If more than 100% is reinvested, it is assumed that the sale of common stock provided the necessary additional funds. A separate and independent (except that the firm is taxed as an entity) project is undertaken each year.

Other Model Parameters

The actual life of a project (L) is an integer random variable between 10 and 20 years. The estimated life of a project is an integer random variable between L-3 and L+3 years. Depreciation schedules for the duration of the actual life are based on the estimated life. For tax purposes, an accelerated depreciation method (sum-of-the-years digits) is used; for reporting purposes, straight line depreciation is used.

The real dollar actual salvage is a random variable ranging between 0 and 20% of the real dollar investment in the project. Let $S_{a,r}$ = actual salvage in real dollars. Let $S_{a,c}$ = actual salvage in current dollars--current meaning dollars as of the original investment. Let $S_{e,c}$ = estimated salvage in current dollars. This is the salvage value used in preparing depreciation schedules for both tax and reporting purposes. $S_{e,c}$ is a random variable ranging between $.5\ S_{a,c}$ and $1.5\ S_{a,c}$.

The before tax real dollar cash return from the project takes one of two basic patterns, selected randomly, each with equal probability. One pattern is basically level over the life of the project. The other is a declining pattern, averaging a 5% reduction per year. Let the real dollar before tax cash flow in year t = R_t. Then:

$$R_{t+1} = R_t(1 - a)\ b$$

Where: $a \begin{cases} = 0 \text{ for level pattern} \\ = .05 \text{ for declining pattern} \end{cases}$

b is a random variable ranging between .917 and 1.083.

R_0, the base used to calculate R_1, is a random variable ranging between A and B times the real dollar investment dividend by the project life. A and B are variables used to induce different degrees of uncertainty and variability into the simulation model. They are discussed later.

The Environment

The environment in which the simulated firm operates, has the following characteristics:

a) Accounting information is prepared on an historical cost basis. Accountants ignore price level changes. Inter-period tax allocations are made by accountants.

b) Inflation averages 3% a year. The price index in year t+1 is a random variable ranging between 1.00 and 1.06 times the price index in year t.

c) Taxes are assessed at a rate of 50% of taxable income. Taxable income is computed in the same manner as accounting income except different depreciation methods are used.

The Variables Manipulate

The variables of interest were inter-year changes in the measured rates of return and the degree of variability embodied in the simulation model. Variability was induced two ways. First, a pseudo random number generator was used to make Monte Carlo draws from specified distributions. The ranges of these distributions have been stated. Their form has not been specified. These distributions took 18 different forms to represent different degrees of uncertainty and variability. Limited uncertainty and variability was represented by symetrical distributions, tightly clustered at the mid-point of the ranges.

Different degrees of variability were represented by spreading out the distributions in steps, through uniform distributions, to U shaped distributions. More specifically, if a continuous random variable had a range from X_* to X^*, then it was selected as

$$X_* + (X^* - X_*)\,\tilde{r}$$

where $\tilde{r}$ is a random variable on the internal 0 to 1. $\tilde{r}$ was based on the beta distribution, calculated to approximate

$$\tilde{r} = F(\tilde{u}|\rho,\nu) \equiv \int_0^{\tilde{u}} f(t|\rho,_\nu)\,dt$$

where:

$$f(t|\rho,_\nu)dt \equiv \frac{t^{\rho-1}\,(1-t)^{\sigma-1}}{\int_0^1 t^{\sigma-1}\,(1-t)^{\sigma-1}dt}$$

$$\sigma = \nu - \rho$$

$$\rho = .5\nu$$

and either

$$\rho \geq 1 \text{ or } \rho,\sigma < 1$$

$\tilde{u}$ was an uniformly distributed random variable from a pseudo random number generator.[7] Since the cumulative beta function cannot generally be evaluated in terms of elementary functions, approximations were necessary. Specifically, $\tilde{r}$ was limited to 100 "equally likely" values for each ν, and Monte Carlo draws were taken from this list. Eighteen selected values of ν, ranging from .05 to 70 were used to induce different degrees of variability and uncertainty.

The second means by which variability and uncertainty was induced was by altering the range of the distribution of base year real dollar before tax cash flow. Previously this range has been referred to as A and B. At one extreme, this range was from 0 to 500% of annualized real dollar cost of the investment. The other extreme was 225% to 275%. Eight intermediate ranges were used, or 10 ranges for base year real dollar before tax cash flow in all. Each of the 10 ranges was used with each of the 18 probability distributions-- resulting in 180 simulation runs, each exhibiting different degrees of variability and uncertainty.

The Concepts of Variability and Uncertainity

It is important that the meaning of the terms variability and uncertainty be clearly stated. Variability merely means that the parameters of individual investment projects differ. No two projects are alike. Each has a different rate of return. Inducing more variability means that the projects tend to display greater differences. Uncertainty means that management does not know all of the parameters of an investment project at the time of its undertaking. Hence true rate of return cannot be calculated on a current basis. Only a surrogate, accounting rate of return can be so provided. There is considerable uncertainty surrounding individual investment projects at the time they are undertaken. Management is represented as not knowing what return would be forthcoming from individual investment projects. In fact, some projects had negative IRR's. Further, the life of a project could only be estimated. Similarly, salvage value is only an estimate. Thus a great deal of uncertainty is expressed at the investment project stage. Since the firm was envisioned as a collection of virtually independent investment projects, some smoothing occurred in aggregation. Because profitable projects were more likely to be undertaken than unprofitable projects, the simulated firm was not prone to bankruptcy. A greater degree of volitability could have been incorporated by allowing investment project parameters to drift over the life of the firm or to create dependencies between returns from existant investments and future investment decisions. Of course, either step might cause instability in the model. The model, as developed, was stable in the sense it produced mean reverting measures of return. None-the-less, the economic outlook for the firm, at any point in time, even tho not explosively unstable, was unknown because it was dependent upon the particular existant and future investment projects, which from the viewpoint of management well all uncertain ventures.

Results

The modeled firm was simulated 180 times, using each of the 18 forms for probability distributions and the 10 ranges described above, in combination. Each run simulated 200 years of firm. The ARR and the TRR were calculated for each year of each simulation. The ARR and TRR measures for one run are shown graphically in Figure 1.

Figure 1 is based on the most volatile situation modeled. This resulted in wide inter-year changes in ARR and TRR. The amount of variability and uncertainty is measured by the standard deviation of the TRR series--in this case .03813. Mere inspection of Figure 1 discloses that the modeled firm's economic performance was highly variable. The efficacy of ARR as a measure of return can be determined by contrasting ARR with TRR. More exactly, inter-year changes in ARR were correlated with inter-year changes in TRR[8]. For the simulation run depicted in Figure 1, the coefficient of correlation between changes in the measures of return is +.77407.

Figure 2 is based on the most minimal state of variability and uncertainty modeled. In this run, the modeled firms economic performance was

Figure I.

Accounting Rate of Return vs. True Rate of Return

Maximum Variability and Uncertainty

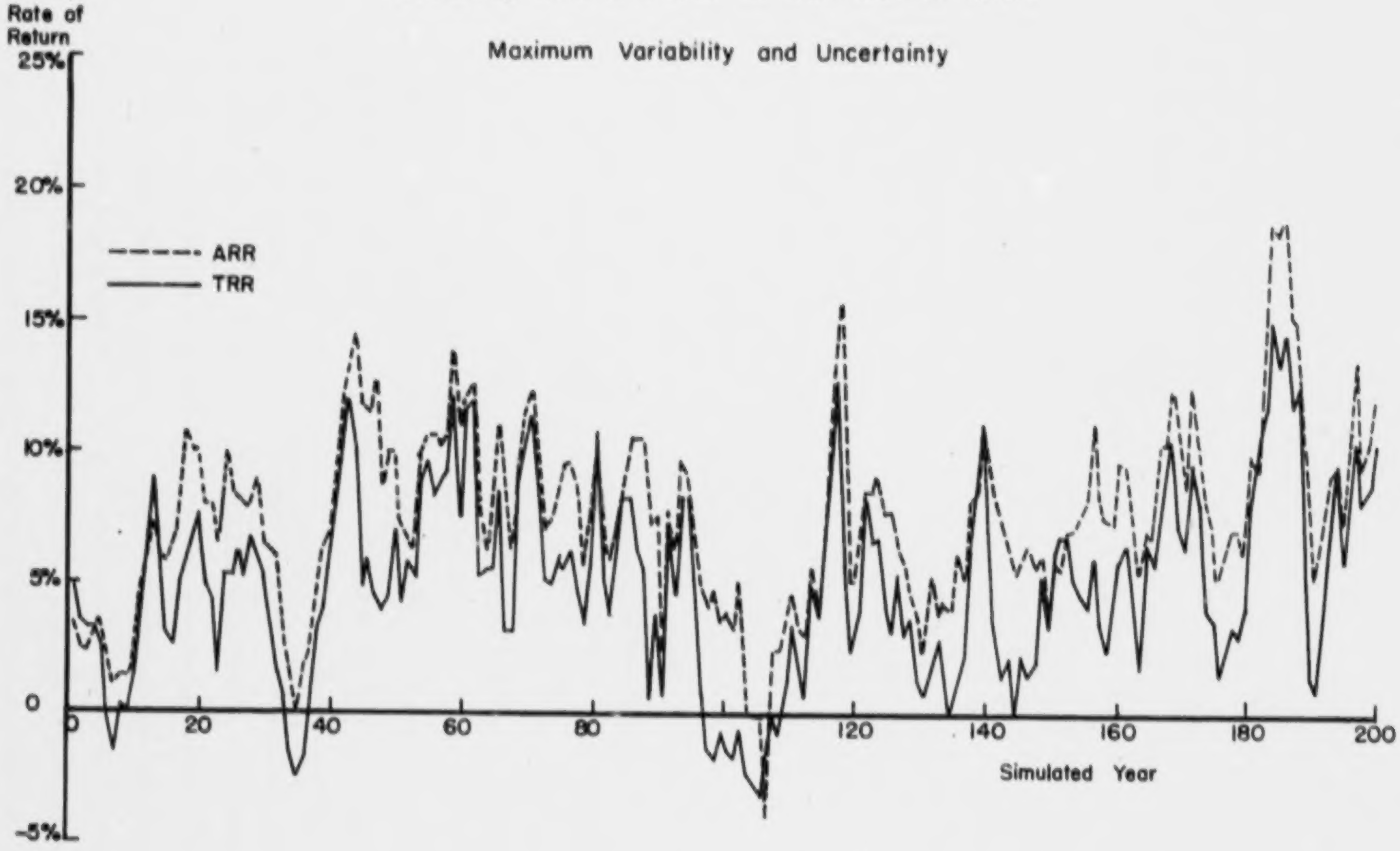
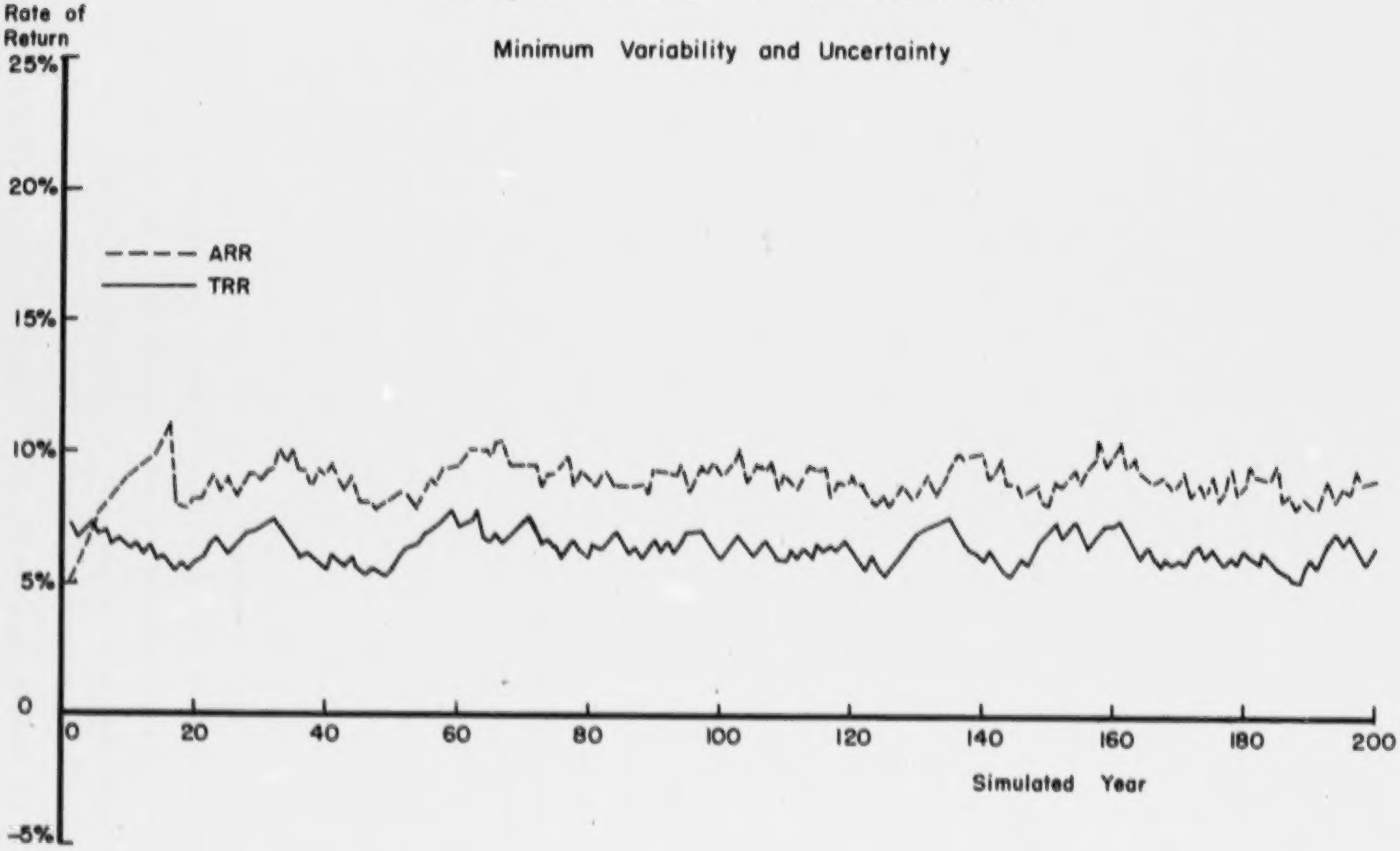


Figure 2.

Accounting Rate of Return vs. True Rate of Return

Minimum Variability and Uncertainty



considerably more stable. As in the case of Figure 1, the ARR is generally greater than the TRR, due in large part to the effects of inflation. However, transient conditions in the early years (roughly years 1 through 20 since the maximum project life was 20 years) were now evident. Therefore, only the last 180 years of the 200 years simulated in this and all other simulations runs were used for results purposes.

The standard deviation of the TRR series in Figure 2 is .00567, far less than .03713 for Figure 1. The coefficient of correlation between changes in ARR and changes in TRR is also far less in Figure 2 than it was in Figure 1; +.11577 vs. +.77407. Thus, the evidence from these two simulation runs is consistent with the crude measuring instrument hypothesis. The more variability and uncertainty, the better accounting measures serve as surrogates for their economic counterparts.

The evidence from all 180 runs is also consistent with the crude measuring instrument hypothesis. Table 4 summarizes all these runs. For each run, in the body of the table, is shown the standard deviation of the TRR series and the coefficient of correlation between inter-year changes in ARR and changes in TRR (the latter being in parentheses). Summary measures previously discussed for the run depicted in Figure 1 are in the upper right-hand corner of the body of Table 4. Figure 2 data is in the lower left-hand corner. The rest of the data in Table 4 is intermediate to the values in these two corners. Table 4 is arranged such that the 18 entries in each column represent runs with constant ranges of base-year real dollar before tax return but different distributional forms for Monte Carlo draws. The 10 entries in each row have constant distributional forms but different ranges for base-year returns. Thus, the two methods for inducing variability and uncertainty form the rows and columns of the table. Generally, moving from left to right across the rows and from bottom to top in the columns means an increase in variability and uncertainty.

Table 4 is analyzed by rows and columns. Consider, for example, the 6th row of Table 4. The ten simulation runs reported in this row all used the same distributional forms for Monte Carlo draws, but different ranges for base-year return. The two concomitant observations for each of the 10 runs in this row are graphically depicted in Figure 3.

Inspection of Figure 3 shows that not only does increasing the range of base-year return increase the standard deviation of the TRR series, but it also increases the coefficient of correlation between inter-year changes in TRR and ARR. The coefficient of correlation between the concomitant observations in the 10 runs is +.93669. This value is shown for row 6 in the margin of Table 4. The correlation of concomitant observations for all the rows and columns are similarly indicated in Table 4. The conclusion being that the runs summarized in each row and column display the interesting relationship that the

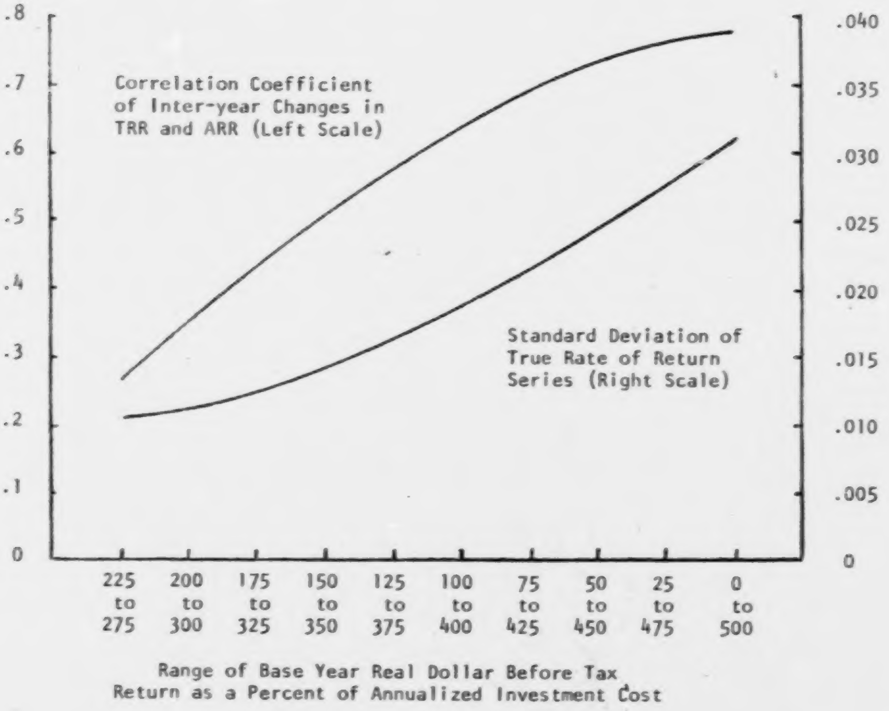
Table 4

Two Concomitant Observations for 180 Simulation Runs: Standard Deviation of True Rate of Return Series and, in Parentheses, Coefficient of Correlation between Inter-year Changes in True Rate of Return: Original parameters.

Distribution Parameter (v)	Range of Base Year Real Dollar Before Tax Return as a Percent of Annualized Investment Cost										Correlation Coefficient of Concomitant Observations in Rows
	225-275	200-300	175-325	150-350	125-375	100-400	75-425	50-450	25-475	0 -500	
.05	.01211 (.26028)	.01286 (.34335)	.01447 (.43058)	.01672 (.52747)	.01940 (.60141)	.02241 (.66001)	.02568 (.70491)	.02919 (.73815)	.03297 (.76125)	.03713 (.77407)	.91870
.10	.01205 (.22673)	.01273 (.31575)	.01424 (.41751)	.01638 (.51118)	.01896 (.58914)	.02187 (.65082)	.02504 (.69815)	.02844 (.73343)	.03211 (.75836)	.03615 (.77312)	.91705
.20	.01200 (.23589)	.01263 (.32566)	.01400 (.42708)	.01611 (.52018)	.01859 (.59766)	.02139 (.65905)	.02446 (.70626)	.02775 (.74156)	.03131 (.76663)	.03522 (.78169)	.91600
.40	.01162 (.25844)	.01226 (.33978)	.01362 (.43422)	.01554 (.52285)	.01787 (.59795)	.02049 (.65831)	.02335 (.70533)	.02643 (.74095)	.02935 (.77673)	.03337 (.78298)	.92379
.60	.01121 (.26034)	.01185 (.33972)	.01319 (.43395)	.01505 (.52331)	.01730 (.59931)	.01984 (.66043)	.02259 (.70802)	.02556 (.74409)	.02875 (.77031)	.03218 (.78720)	.92690
.80	.01094 (.27465)	.01155 (.34299)	.01282 (.42889)	.01460 (.51351)	.01675 (.58761)	.01917 (.64862)	.02181 (.69711)	.02464 (.73462)	.02768 (.76261)	.03096 (.78167)	.93669
2	.00923 (.07642)	.00964 (.12890)	.01052 (.21083)	.01178 (.30340)	.01333 (.39357)	.01509 (.47473)	.01702 (.54455)	.01910 (.60300)	.02130 (.65104)	.02365 (.68784)	.96623
5	.00740 (.17897)	.00762 (.22639)	.00814 (.29155)	.00889 (.36382)	.00984 (.43525)	.01093 (.50125)	.01215 (.55979)	.01345 (.61045)	.01483 (.65360)	.01626 (.68796)	.96842
10	.00608 (.09700)	.00619 (.12820)	.00648 (.17514)	.00693 (.23199)	.00752 (.29292)	.00822 (.35392)	.00901 (.41211)	.00987 (.46592)	.01078 (.51463)	.01173 (.55812)	.98181
15	.00603 (.17476)	.00607 (.19198)	.00624 (.22136)	.00653 (.25983)	.00693 (.30404)	.00742 (.35097)	.00799 (.39822)	.00862 (.44410)	.00931 (.48750)	.01003 (.52781)	.98710
20	.00585 (.15778)	.00585 (.17448)	.00594 (.20068)	.00614 (.23429)	.00643 (.27296)	.00680 (.31447)	.00723 (.35695)	.00773 (.39878)	.00827 (.43955)	.00866 (.47801)	.98273
25	.00586 (.14399)	.00587 (.16287)	.00595 (.18918)	.00612 (.22125)	.00636 (.25731)	.00666 (.29566)	.00702 (.33427)	.00743 (.37381)	.00789 (.41166)	.00838 (.44786)	.98132
30	.00577 (.12879)	.00578 (.14930)	.00586 (.17059)	.00600 (.20034)	.00621 (.23406)	.00648 (.27034)	.00679 (.30788)	.00715 (.34565)	.00755 (.38281)	.00798 (.41878)	.98393
35	.00576 (.16581)	.00576 (.18369)	.00582 (.20704)	.00594 (.23477)	.00612 (.26570)	.00635 (.29862)	.00663 (.33268)	.00694 (.36685)	.00730 (.40053)	.00768 (.43322)	.97968
40	.00574 (.17629)	.00576 (.19166)	.00582 (.21229)	.00594 (.23725)	.00610 (.26552)	.00631 (.29605)	.00656 (.32790)	.00685 (.36025)	.00717 (.39342)	.00752 (.42391)	.98453
50	.00576 (.11331)	.00575 (.12064)	.00578 (.13311)	.00588 (.15021)	.00598 (.17127)	.00613 (.19553)	.00632 (.22224)	.00655 (.25065)	.00680 (.28010)	.00709 (.31001)	.98957
60	.00568 (.11243)	.00568 (.11832)	.00572 (.12977)	.00579 (.14341)	.00589 (.16174)	.00603 (.18317)	.00620 (.20709)	.00640 (.23268)	.00662 (.25996)	.00687 (.28782)	.99354
70	.00567 (.11577)	.00567 (.11844)	.00571 (.12504)	.00577 (.13536)	.00586 (.14906)	.00598 (.16574)	.00613 (.18496)	.00630 (.20625)	.00649 (.22915)	.00671 (.25321)	.99750
Correlation Coefficient of Concomitant Observations in Columns	.78393	.89574	.94587	.96250	.96566	.96253	.95596	.94706	.93580	.92142	

Figure 3

Graphical Depiction of Concomitant Observations Reported in Row 6 of Table 4



greater the degree of variability and uncertainty, the better accounting measures serve as surrogates for economic concepts.[9]

A Macro-Sensitivity Analysis

The parameters of the model were selected in what can only be called an arbitrary manner. The only defense that can be made for the particular values selected is that they were thought to be reasonable. Other values would also be reasonable. As a gross test of sensitivity of the model to parameter values, a new set of values were selected and the complete study rerun. The original and new parameter values are listed in Table 5. Results, using new values, are shown in Table 6.

It should be noted that in this version of the simulation, the single project per year requirement is removed, adding greatly to the computing time required.

Variability and uncertainty were again manipulated by altering the forms of probability distributions and the ranges of base-year return. Results using new parameters, as shown in Table 6, were similar to those previously obtained. Again, the greater the degree of variability and uncertainty in the ARR series, the higher the correlation between inter-year changes in the two return measurers.[10] Consistent results from this second version of the simulation allows at least a minor degree of confidence that there is some generality to the relationships found, but of course caution is warranted. At least the results are not dependent upon the original parameter values selected. Two sets of reasonable parameter values produced consistent results.

Discussion

This paper has treated accounting measures of return as crude surrogates for an economic concept. A method for assessing how crude the surrogate is has been developed, but more importantly, the Monte Carlo simulation technique has been brought to bear on an area of inquiry which has heretofore seen almost exclusive reliance upon certainty models.[11] One point that must be stated strongly is that certainty models are probably inadequate. The degree of environmental variability and uncertainty has been shown to be an important factor when contrasting accounting and economic measures.

Finally, some potential areas for further research can be identified. For example, one might ask if adjustments for price level changes in the accounts, as suggested by the accounting profession[12], would improve or detract from the correlation measures used in this study? Lead-log relationships and moving averages could be investigated. Economic cycles could be introduced into cash flow patterns. Thus, a host of additional research opportunities present themselves.

TABLE 5

Parameters Used in Simulation Model

Parameter	Original	New
Percent of available cash flow re-invested		
Maximum	150%	100%
Minimum	50%	50%
Number of investment projects undertaken annually	1	3
Cost of an investment as a proportion of current capital budget	100%	18.3 to 50%
Life of investment project		
Maximum	20 years	15 years
Minimum	10 years	5 years
Accountant's error in estimating life of investment project		
Maximum over estimate	3 years	4 years
Maximum under estimate	3 years	1 year
Actual salvage value in real dollars as a proportion of investment		
Maximum	20%	30%
Minimum	0%	10%
Accountant's error in estimating salvage		
Maximum over estimation	50%	25%
Maximum under estimation	50%	25%
Patterns of cash returns from investment		
Probability of level returns	50%	80%
Probability of decreasing returns	50%	20%
Average annual reduction with decreasing returns	5%	7%
Random deviations from pattern (annual)		
Maximum	8.3%	15%
Minimum	8.3%	10%
Price level changes (annual)	+6%	+10%
Maximum	0%	+ 2%
Income tax rate	50%	40%

Table 6

Two Concomitant Observations for 180 Simulation Runs: Standard Deviation of True Rate of Return Series and, in Parentheses, Coefficient of Correlation between Inter-year Changes in Accounting Rate of Return and Inter-year Changes in True Rate of Return: New Parameters.

Distribution Parameter (ν)	Range of Base Year Real Dollar Before Tax Return as a Percent of Annualized Investment Cost										Correlation Coefficient of Concomitant Observations in Rows
	225-275	200-300	175-325	150-350	125-375	100-400	75-425	50-450	25-475	0 -500	
.05	.01743 (.49420)	.01971 (.52079)	.02310 (.56307)	.02720 (.61083)	.03174 (.65693)	.03660 (.69777)	.04170 (.73213)	.04700 (.76000)	.05250 (.78185)	.05822 (.79811)	.97731
.10	.01705 (.48295)	.01925 (.50408)	.02257 (.54394)	.02661 (.59154)	.03111 (.63875)	.03593 (.68122)	.04098 (.71729)	.04623 (.74678)	.05167 (.77004)	.05733 (.78748)	.98100
.20	.01651 (.46683)	.01877 (.48217)	.02213 (.51792)	.02616 (.56308)	.03063 (.60925)	.03539 (.65151)	.04038 (.68802)	.04554 (.71826)	.05089 (.74243)	.05645 (.76079)	.98565
.40	.01636 (.39543)	.01845 (.41714)	.02156 (.46014)	.02532 (.51238)	.02951 (.56508)	.03400 (.61342)	.03871 (.65543)	.04360 (.69271)	.04867 (.71954)	.05394 (.74239)	.98523
.60	.01563 (.39330)	.01762 (.41369)	.02055 (.45327)	.02410 (.50188)	.02805 (.55199)	.03227 (.59912)	.03670 (.64113)	.04130 (.67732)	.04606 (.70771)	.05101 (.73259)	.98894
.80	.01509 (.45752)	.01710 (.48033)	.01998 (.51871)	.02343 (.56375)	.02724 (.60894)	.03131 (.65059)	.03556 (.68713)	.03998 (.71816)	.04454 (.74389)	.04926 (.76469)	.98625
2	.01223 (.44656)	.01373 (.46024)	.01581 (.48822)	.01827 (.52458)	.02100 (.56411)	.02390 (.60317)	.02694 (.63967)	.03010 (.67263)	.03335 (.70164)	.03470 (.72684)	.99536
5	.00957 (.33163)	.01040 (.35165)	.01159 (.38252)	.01304 (.42000)	.01467 (.46047)	.01644 (.50128)	.01831 (.54071)	.02025 (.57775)	.02226 (.61186)	.02432 (.64278)	.99618
10	.00822 (.31931)	.00880 (.34423)	.00960 (.37702)	.01058 (.41429)	.01168 (.45336)	.01288 (.49229)	.01416 (.52984)	.01553 (.56523)	.01689 (.59805)	.01832 (.62812)	.99426
15	.00761 (.25036)	.00804 (.27120)	.00865 (.29877)	.00940 (.33060)	.01026 (.36465)	.01121 (.39941)	.01223 (.43384)	.01330 (.46726)	.01441 (.49922)	.01556 (.52945)	.99607
20	.00733 (.30738)	.00768 (.31935)	.00817 (.33664)	.00878 (.35773)	.00949 (.38145)	.01028 (.40681)	.01112 (.43296)	.01202 (.45927)	.01296 (.48525)	.01393 (.51055)	.99883
25	.00692 (.30830)	.00723 (.32107)	.00765 (.33818)	.00817 (.35862)	.00878 (.38139)	.00946 (.40564)	.01019 (.43067)	.01097 (.45591)	.01178 (.48094)	.01263 (.50542)	.99849
30	.00644 (.33058)	.00670 (.34753)	.00708 (.36766)	.00754 (.38995)	.00809 (.41357)	.00870 (.43784)	.00936 (.46228)	.01007 (.48649)	.01081 (.51018)	.01158 (.53314)	.99500
35	.00619 (.28894)	.00645 (.30740)	.00680 (.32881)	.00724 (.35209)	.00775 (.37640)	.00832 (.40109)	.00893 (.42572)	.00959 (.44998)	.01027 (.47365)	.01099 (.49658)	.99415
40	.00597 (.25649)	.00619 (.27373)	.00650 (.29470)	.00689 (.31824)	.00735 (.34333)	.00786 (.36916)	.00842 (.39515)	.00902 (.42086)	.00966 (.44600)	.01031 (.47039)	.99498
50	.00598 (.23447)	.00618 (.24822)	.00645 (.26534)	.00678 (.28499)	.00718 (.30638)	.00762 (.32885)	.00811 (.35186)	.00863 (.37503)	.00918 (.39805)	.00975 (.42071)	.99698
60	.00605 (.26191)	.00622 (.27533)	.00645 (.29134)	.00673 (.30937)	.00707 (.32876)	.00745 (.34901)	.00787 (.36969)	.00832 (.39050)	.00880 (.41118)	.00930 (.43157)	.99585
70	.00607 (.22039)	.00623 (.23368)	.00643 (.25013)	.00669 (.26904)	.00699 (.28973)	.00732 (.31158)	.00769 (.33407)	.00809 (.35679)	.00852 (.37943)	.00897 (.40176)	.99697
Correlation Coefficient of Concomitant Observations in Columns	.92057	.92692	.94083	.95258	.95951	.96235	.96219	.95981	.95561	.94965	

Notes and References

1 A recent monograph on this subject contains much of the previously reported research. J. Leslie Livingstone and Thomas J. Burns, Income Theory and Rate of Return, College of Administrative Science Monograph S-2, The Ohio State University, 1971. An excellent survey article, also contained in the above monograph, is John Leslie Livingstone and Gerald L. Salamon, "Relationship Between the Accounting and Internal Rate of Return Measures: A Synthesis and An Analysis," Journal of Accounting Research, Vol. 8, No. 2, Autumn, 1970, pp. 199-216. A more recent, and perhaps the most comprehensive study to date is Thomas R. Stauffer, "The Measurement of Corporate Rates of Return: A Generalized Formulation," The Bell Journal of Economics and Management Science, Vol. 2, Autumn, 1971, pp. 434-469.

2 See Melvin N. Greenball, "The Accuracy of Different Methods of Accounting for Earnings - A Simulation Approach," Journal of Accounting Research, Vol. 6, No. 1, Spring, 1968, pp. 114-129, for a different model formulation.

3 Often referred to as the compound-interest method of amortization.

4 Of course the TRR could not be calculated until all of the required ex post information is known.

5 The capital recovery (or compound-interest) method requires an interest rate. The project IRR is used for the assets of each project. It should be noted that the use of project IRR's rather than a firm's cost of capital has been criticized in William J. Vatter, "Income Models, Book Yield and Rate of Return," The Accounting Review, Vol. 41, No. 4, October, 1966, pp. 681-698.

6 Accounting Principles Board Opinion Number 7, May, 1966.

7 The pseudo-random number generator was derived from P. A. W. Lewis, A. A. Goodman, and J. M. Miller, "A Pseudo-Random Number Generator for the System/360," IBM Systems Journal, Vol. 8, No. 2, 1969, pp. 136-146.

8 Evaluating the information content of changes in ARR rather than of ARR itself reflects the dominant role of the consistency convention in Accounting practice.

9 I would prefer to limit statiscal analysis to descriptive statistics since the assumptions of classical statistical methods are not met in such a computer simulation experiment. See R. W. Conway, "Some Tactical Problems in Simulation Method," Memorandum RM-3244-PR, The Rand Corporation, Santa Monica, California, October, 1962, as reported in Claude McMillan and Richard F. Gonzalez, Systems Analysis: A Computer Approach to Decision Models. Homewood, Illinois: Richard D. Irwin, Inc., 1965. However, there seems to be interest in tests of inappropriate null-hypotheses. In this vain, the following statistics are offered. For the 18 row r's, tests to reject the null hypotheses that r=0 resulted in t's ranging from 6.458 to 39.942; each with 8 degrees of freedom. For the 10 columns, t ranged from 5.050 to 14.866; each with 16 degrees of freedom.

10 For the 18 row r's, tests to reject the null hypotheses that r=0 resulted in t's ranging from 13.049 to 58.386; each with 8 degrees of freedom. For the 10 columns, t ranged from 9.880 to 14.161; each with 16 degrees of freedom! See Note 9.

11 Livingstone and Salamon op. cit. concluded their paper by noting "Like other studies cited, this one also assumes a constant IRR for all projects. While convenient, this assumption is certainly limiting and its removal would provide a better approximation to the real world. So, likewise, would the recognition of uncertainty."

12 See Accounting Principles Board Statement No. 3, "Financial Statements Restated for General Price-Level Changes," June, 1969.

VARIABILITY ASSUMPTIONS AND THEIR EFFECT ON CAPITAL INVESTMENT RISK

by

F. J. Brewerton
Louisiana Tech University

William B. Allen
United States Air Force

Introduction

Conventional methods of analyzing risk in capital investment decisions fail to represent investment risk or variability in investment return because the calculation of rates of return using simple averages of the basic investment factors disguises the combined effect of external factor values on the rate of return. With Monte Carlo simulation, the sampling of investment factors according to their probability estimates allows the representation of factor variability in the rate of return calculations.

The usual procedure of assuming normally distributed factor probability is a simplifying assumption that influences the representation of variability and risk. According to statistical theory, Monte Carlo simulations with non-normal variability assumptions will produce simulated rates of return that are normally distributed but with greater degrees of variability. The purpose of this paper is to examine the results of simulations which utilize non-normal factors and to compare these results with those obtained using normally distributed investment factors and mean value investment factors.

A simulation approach rather than an analytical approach is used in the study because of the nature of the bounded statistical distributions used in the investment model. The model itself is complex to the extent that analytically handling the investment factors included in the model generates considerable mathematical tedium. Furthermore, the mathematics of combining statistical distributions of varying types can be extremely difficult and at times impossible.

Assumptions and Limitations

The scope of this study is restricted to those decisions involving one particular category of capital investment, the investment in initial production facilities. The type of investment under study is defined as a single cash outlay which produces varying revenues at varying costs over a future useful life. The study is further limited to a particular business environment in which the investment is influenced only by specified factors. These factors include the major marketing, investment, and production variables that contribute to the determination of profitability. It is further assumed that there is perfect positive correlation between the rate of return for this type of investment and that of the firm as a whole. Not included are such considerations as the opportunity for alternative investments, the cost of financing, or limitations on the amount of financing available. In general, the hypothetical investment decision included in this study is concerned only with the profitability of a single capital investment measured solely by its own earnings.

Other basic assumptions surrounding the study include:

(1) A reasonable approximation of risk is acceptable for confident, effective decision-making. Although other measures of risk are available, the standard deviation of the rate of return distribution is considered an acceptable measure of risk and will be used to measure risk in this study.

(2) Techniques of sales forecasting, cost projection, and investment prediction provide subjective probability estimates which are statistically valid.

(3) Computer programming and processing facilities are available at reasonable cost. (These costs are not included in the factors included in the investment model.)

(4) Monte Carlo simulation of the investment model reasonably approximates actual behavior in decision-making.

(5) The simulation model reasonably represents the pertinent factors and relationships of the investment decision.

(6) The internal rate of return is a reasonable measure of an investment's attractiveness.

Factors Influencing Investment Profitability

The analysis of risk in a proposed capital investment requires identification of the basic investment factors contributing to the determination of profitability and which have a significant effect on the risk of achieving expected profitability. Since the future values for investment factors may have different values than estimated, the final return on investment is subject to considerable variability. The key variable factors seem to be those that directly relate to investment earnings, such as sales revenue and production costs, or those that directly influence the nature of the investment, such as the amount and life of the investment.

In actual applications of risk analysis, the choice of significant factors will depend upon the particular market, production, and investment characteristics. For example, Hertz [1] selected as the key factors such variables as market size, selling prices, market growth

500

rate, share of market, investment required, residual value of investment, operating costs, fixed costs, and useful life of facilities. Wagle [5] selected the same factors while Hillier [3] considered only revenue and cash flows. Hess and Quigley [2] used demand, price, fixed cost, variable cost, amount of investment, and plant capacity.

The simulation model here identifies and utilizes eleven investment factors as being significant. The selection of these particular factors results from the desire to construct a hypothetical investment that involves a relatively high level of risk. The factors are grouped into four classes and are defined as follows:

I. Marketing Factors

- A. PRCMKT - the dollar sales price of the investment product.
- B. VOLMKT - the number of product units of total industry sales.
- C. SHARE - the firm's percentage share of the total industry sales.

II. Production Factors

- A. VARCST - the variable manufacturing costs in dollars per unit, including taxes.
- B. FXDCST - the dollar amount of fixed manufacturing cost, including depreciation.
- C. SAEXP - the dollar amount of selling and administrative expenses.

III. Investment Factors

- A. RQDINV - the dollar amount of investment required for beginning production.
- B. INVLFE - the useful production life of the investment.
- C. SALVAG - the residual value of the investment at the end of its useful life.

IV. Dynamic Factors

- A. GROWTH - the percentage rate of change in market sales volume.
- B. RINFLA - the percentage rate of change in product prices and production costs and expenses.

The Hypothetical Investment Model

The hypothetical model used in this study to demonstrate Monte Carlo simulation and to evaluate different factor variability assumptions is constructed primarialy with the aim of revealing the risk involved in typical capital investment decisions. Emphasis in the model is on the number of probabilistic investment factors and the variability of their probability estimates. This emphasis allows the element of risk to be reflected in the rate of investment return without undue distortion from other sources.

Although basically an artificial construct, the hypothetical model is designed to represent reasonably realistic business conditions but not to parallel any particular capital investment. Actual applications of Monte Carlo simulation will necessitate a specific model tailored to the particular investment situation. Essential elements of the model are the rate of return function, the basic investment factors of the return function, the interrelationships between these factors, the estimated numerical values for the factors, and the type of variability in the factors.

The Rate of Return Function

Measuring profitability by the discounted cash flow method determines a rate of return which equates the sum of the present values of future period cash flows to the amount of initial investment. It is defined as that rate of return r which equates the initial investment I to the sum of expected cash flows C_1, C_2, ..., C_n as follows,

$$I = C_1 \left[\frac{1}{(1+r)^1}\right] + C_2 \left[\frac{1}{(1+r)^2}\right] + \ldots + C_n \left[\frac{1}{(1+r)^n}\right] \qquad (1)$$

in which the value of r is found by trial and error techniques [6].

Since the desired simulation model is one which represents the objective function, the hypothetical investment model is essentially the discounted rate of return equation. Expressed in the programming notation used for the computer simulation, the equation is:

RQDINV = PRVALU, in which

$$\text{PRVALU} = \sum_{j=1}^{\text{INVLFE}} \text{CASHFL}(j) \left[\frac{1}{(1 + \text{RATE})^j}\right] + \text{SALVAG} \left[\frac{1}{(1 + \text{RATE})^{\text{INVLFE}}}\right] \qquad (2)$$

In the above relationship, RQDINV is the amount of initial investment required, PRVALU represents the present value of future cash flows, CASHFL(j) is the expected net cash receipts in period j, RATE is the rate of return which causes equality, and INVLFE is the useful life of the investment.

The net cash flow, CASHFL(j), consists of the sum of cash inflows less cash outflows. According to the discounted cash flow method, cash inflows include the sales revenue each period plus depreciation charges. Cash outflows are the costs of owning and operating the investment each period. The model computes the rate of return after taxes, with period taxes being treated as part of period costs. Considering the model equation with these component cash flows, then

$$\text{CASHFL}(j) = \text{CFSALE}_j - \text{CFCOST}_j + \text{DEPREC}_j \qquad (3)$$

in which CFSALE represents sales revenue each period, CFCOST is the total accounting cost in each period, and DEPREC refers to the depreciation charge for each period. The expression $(- \text{CFCOST}_j + \text{DEPREC}_j)$ is equivalent to the cash outflows per period and CFSALE (Sales revenue) is equivalent to cash inflows per period.

It is possible (and perhaps more desirable) to compute the period cash flows by subtracting period expenses from sales revenues and thus eliminating depreciation as a variable in the model. Such an approach presupposes that the cash flow information is available. Accounting records very often are the only source of input information for the model despite the fact that accounting "costs" may not be identical to "expenses." In such instances the accounting information must be properly adjusted to coincide with actual cash flows. For example, the two production variables FMFCST and VMFCST contain elements of depreciation which are not cash flow elements. The model thus recognizes the disparity between accounting costs and expenses and determines the period net cash flows in the manner indicated in equation (3).

The model uses the straight-line method of depreciation accounting for simplicity. Actual simulations would follow the particular convention of the user. By the straight-line method, the depreciation charge each period is the net amount of investment divided by the expected useful life of the investment. The equation for the model is

$$\text{DEPREC}_j = (\text{RQDINV} - \text{SALVAG}) / \text{INVLFE}. \qquad (4)$$

Other cash flows in the model are defined as

$$\text{CFSALE} = \text{PRCMKT} * \text{SALVOL}, \qquad (5)$$

in which

$$\text{SALVOL} = \text{SHARE} * \text{VOLMKT} \qquad (6)$$

and

$$\text{CFCOST} = \text{FSDCST} + (\text{VARCST} * \text{SALVOL}) + \text{SAEXP} \quad (7)$$

The model assumes that any time lags are constant over the life of the investment, and consequently the effect of time lags on rate of return variability is ignored for simplicity. Following this assumption, the investment begins production simultaneously with the investment, and sales occur at the time and rate of production.

The model recognizes two sets of functionally correlated investment factors, market price-sales volume and sales volume-variable manufacturing costs. Statistical interrelationship between these correlated factors is accomplished by utilizing multiple subjective probability estimates. For each possible value of one functionally correlated variable, there is a range of possible values for the other variable. These multiple estimates represent each factor as statistically independent although the variables themselves are functionally correlated.

Each time period of the investment life is also interrelated. To maintain statistical independence and to allow for functional correlation over time, the dynamic factors of growth and inflation serve to relate factors in earlier periods to the present simulation period. The model is constructed, and the simulation is performed so that the rates of growth and inflation determine a new range for the probability distribution of each investment factor.

To illustrate the sensitivity of rate of return variability to the assumption of the type of factor probability distribution, separate simulations of the hypothetical investment were performed using six different assumptions. They are a bounded standard normal distribution, a bounded peaked normal distribution, a bounded flat normal distribution, a bounded left-skewed distribution, a bounded right-skewed distribution, and a random selection of these five distributions.

These distributions are defined by varying the parameters of two basic functions. The normal function is symmetrical, and different standard deviations change the degree of peakedness for a given mean. Its equation is [4],

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \quad e^{-\frac{1}{2}\left[\frac{x-\bar{x}}{\sigma}\right]^2} \qquad (8)$$

The Beta function is skewed, and different parameters change both peakedness and skewness. Its equation is [4],

$$f(x) = \frac{(a+b+1)\;!}{a\;!\;b\;!}\; x^a\;(1-x)^b \qquad (9)$$

Figure 1 depicts the five types of distributions over the unit interval. The selection of these distributions is arbitrary and is intended only to reveal moderate departures from normal variability. Having assumed a type of variability, numerical parameters are necessary to completely define the factor subjective probability distributions.

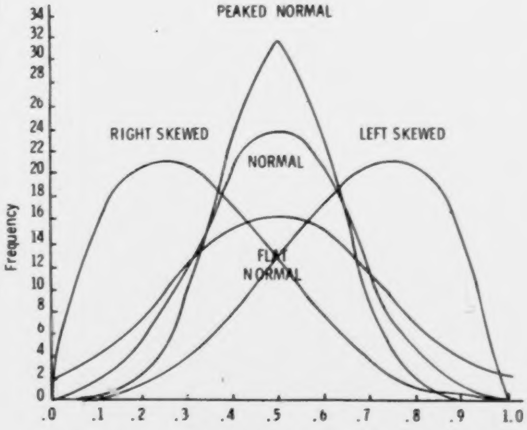


Figure 1 The Five Distributions of the Simulation Over the Unit Interval

Numerical Factor Values for the Model

The extremes of a distribution are a means to quantify and bound a function when the type of function between these extremes is known. Accordingly, simulation of the model utilizes the extreme high and low range estimates in conjunction with the assumed type of variability to provide subjective probability estimates for each investment factor.

The numerical values for the range estimates (bounds) of the eleven investment factors are arbitrary. The objective in the model is only to produce a measurable rate of return and to reveal the effect of factor variability upon rate of return variability. Whenever reference is made to any of the distributions used in the study, it is understood that they are bounded distributions and do not necessarily correspond precisely to their theoretical counterparts.

Simulation Flow Diagram

The logic flow diagram for simulating the hypothetical investment is shown in Figure 2. The basic flow is to simulate randomly different investment lives from the same conditions and to compute the return for each life. After all simulations, probabilities and statistical measures are calculated for the returns simulated, then, for comparison, the rate of return is computed by averaging factor estimates.

Simulation of an investment life first involves a definition of the hypothetical investment by determining the amount of investment (RQDINV), the life of the investment (INVLFE), and its residual value (SALVAG). Values for these factors, like all eleven factors, are selected by the Monte Carlo method. A value is randomly selected between the high and low range estimates according to the type of probability assumed for the investment. After this definition of the investment, each period of investment life is simulated.

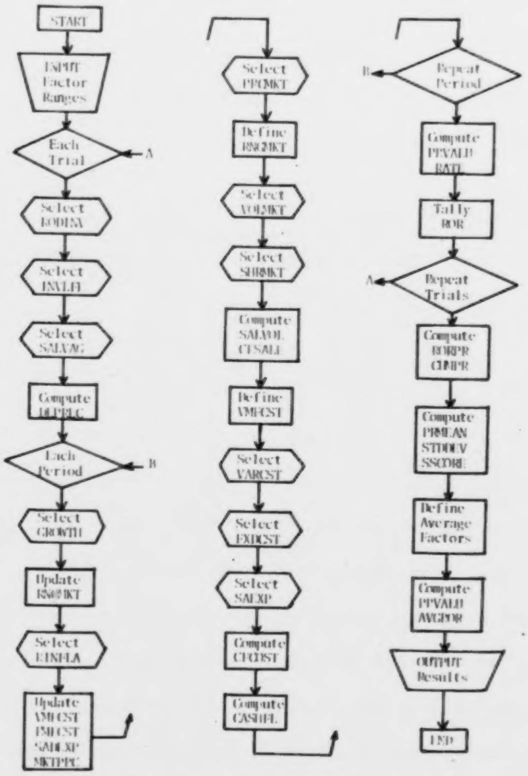

START
INPUT Factor Ranges
Each Trial
A
Select RQDINV
Select INVLFE
Select SALVAG
Compute DEPREC
Each Period
B
Select GROWTH
Update RNGMKT
Select RINFLA
Update VMFCST FMFCST SAEXP MKTPPC
Select PRCMKT
Define RNGMKT
Select VOLMKT
Select SHRMKT
Compute SALVOL CFSALE
Define VMFCST
Select VARCST
Select FXDCST
Select SAEXP
Compute CFCOST
Compute CASHFL
B
Repeat Period
Compute PPVALU RATE
Tally ROR
A
Repeat Trials
Compute RORPR CUMPR
Compute PRMEAN STDDEV SSCORE
Define Average Factors
Compute PPVALU AVGROR
OUTPUT Results
END

Figure 2 Program Flow Diagram

The rate of growth in market volume (GROWTH) and the rate of price-cost increase (RINFLA) are selected at the start of each period except the first. These rates update the ranges of market volume (RNGMKT) and the three costs. Values for the factors determining revenue cash flow are determined next. The selection of a market price (PRCMKT) also defines one of fifteen possible market ranges (RNGMKT) allowing for price-volume correlation. A value of market volume (VOLMKT) is then selected from the defined market range. After selecting a value for the firm's share of this market volume (SHRMKT), the sales volume (SALVOL) is computed as the product of share and volume. The revenue cash flow (CFSALE) is then computed from price and volume.

The next phase consists of selecting values [illegible]r cash outflows. The value of sales volume, representing production volume, defines one of ten ranges for the correlated variable cost (VMFCST). After selecting this cost (VARCST), values for fixed manufacturing (FXDCST) and selling-administrative expenses (SAEXP) are selected. Cash outflow (CFCOST) is calculated by summing these costs and expenses. The net cash flow (CASHFL) is then calculated from the revenue, cost, and depreciation cash flows. This sequence is repeated for each period of the investment life.

When all periods have been simulated, the rate of return is computed from the period net cash flows and the residual value of the investment. The calculated rate of return is recorded, and the entire simulation is repeated for another trial simulating another investment with the same set of investment conditions but continuing to randomly select factor values.

After the desired number of investment trials has been simulated, statistical and probability measures are calculated for the returns simulated to provide an indication of the approximate risk in the hypothetical investment. The rate of return is computed using the conventional risk analysis method of averaging factor values for a comparison. First, the average value for each of the eleven investment factors is calculated from their high and low estimates. From these averages, the cash flows are computed for the average life of the investment. The rate of return is then determined by the same procedure as in the simulation.

Results of The Model Simulation

The computer simulation of the hypothetical investment produced a distribution of rates of return as the objective measure of risk in the proposed investment. The important results are:

1. The simulation using the assumption of normal variability in the investment factor estimates produced a distribution of returns with an expected rate of return which was much higher than the rate of return computed from the average of factor values.
2. The simulation that randomly used the five types of variability resulted in a distribution of returns with a lower expected value and a larger variation than obtained by the normal variability simulation.
3. The simulations involving four types of factor variability, which were non-normal, gave rates of return that were significantly different from the distribution of returns from the normal variability simulation. These distributions also directly reflected the type of variability distribution assumed for the factors.

These results suggest that, when measuring risk by the variability of the simulated rates of return, Monte Carlo simulation gives a different representation of risk depending upon the assumption of variability in the basic investment factors. These results further imply that the expected rate of return approximated by Monte Carlo simulation is likely to differ from the expected return computed by the conventional method of risk analysis.

The Rates of Return Simulated

The rates of return obtained by the computer simulation are presented in Figures 3 and 4. Figure 3 involves 1000 trial frequency distributions while Figure 4 involves 1600 trials. The curve for the simulation of normal variability conditions in Figure 3 is based upon the first 1000 trials whereas in Figure 4 the curve is based upon the total 1600 trials. The significance of these Figures is that they reveal how the distribution of returns varied over a range of possible returns from 0 to 270 percent.

Referring to Figure 3, the distribution for the five simulations varied over the range of returns according to the type of investment factor variability assumed during the simulation. The simulation assuming normal variability produced a distribution that was approximately symmetrical over the range of returns. The distribution for the simulation assuming peaked normal factor variability resulted in a distribution that was also symmetrical about the normal distribution but was more peaked than the normal. The distribution for the flat normal simulation resulted in a distribution that was also symmetrical but less peaked than the normal. The distributions for the right and left skewed simulations produced distributions that were skewed right and left respectively from the normal distribution with peakedness similar to the normal distribution. The character of these five curves corresponds very closely to the character of the five curves of the basic types of factor variability shown earlier in Figure 1.

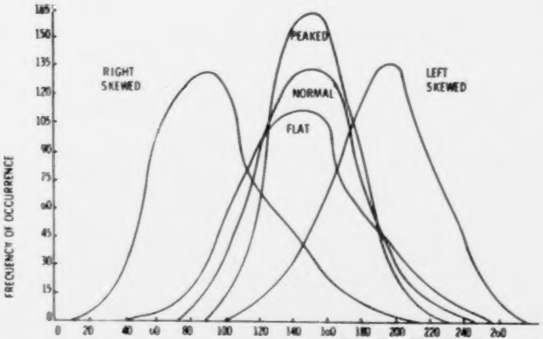


Figure 3 Simulation Rate of Return Frequency Distributions for 1000 Trials/Assuming Standard Normal, Peaked Normal, Flat Normal, Left-skewed, and Right-skewed Types of Investment Factor Variability

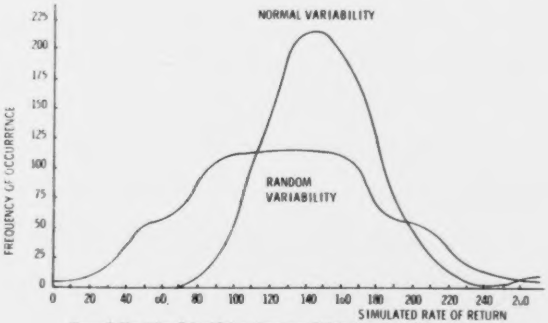


Figure 4 Simulation Rate of Return Frequency Distributions for 1600 Trials/Assuming Normal Variability and a Random Selection of Five Types of Variability

Figure 4 shows the rate of return frequency curves for the simulations of 1600 trials using normal variability and a random selection of the five types of variability. The random variability distribution is clearly different from the normal distribution. The random curve is much more variable than the normal curve and is symmetrical at a much lower rate of return than the normal curve. Statistical measures provide a numerical description of the distribution characteristics.

The simulation also produced a distribution of simulated returns and computed a single rate of return by the conventional averaging method. The comparative advantage is shown in Figures 5 and 6. Figure 5 represents the "risk profile" of

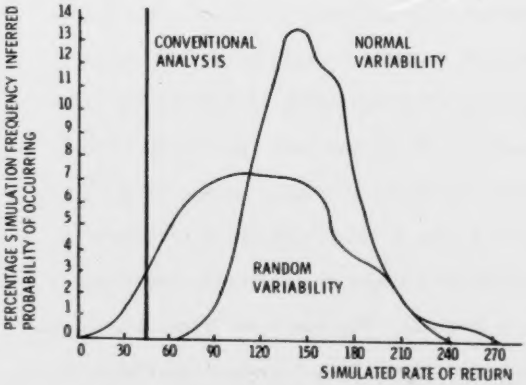


Figure 5 Risk Profile of Hypothetical Investment

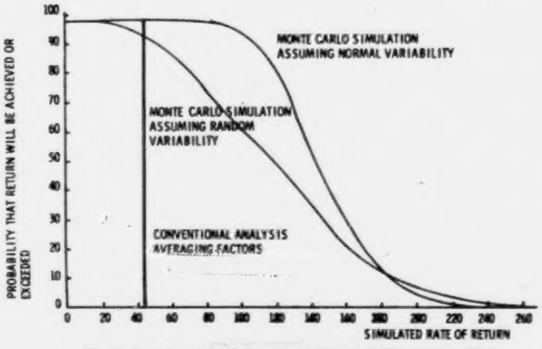


Figure 6 Comparison of Expected Investment Return Probabilities for Conventional and Monte Carlo Methods

the hypothetical investment for the conditions of normal and random factor variability. It is a percentage frequency distribution of the various simulated returns and indicates not only the variability in investment return but also the inferred probability of each rate of return. Figure 5 also includes the single expected rate of return computed by the averaging method. For the hypothetical investment, a significant difference in the anticipated investment returns is apparent.

Figure 6 shows the cumulative probability curves for the two conditions of investment and indicates the probability of realizing higher rates of return. It was drawn from cumulative probability values calculated from simulation results. The probability for the investment return using the averaging method is one since it is a single value. Figure 6 reveals a difference between the averaging and Monte Carlo methods. Figures 5 and 6 are also representative of the decision guide available to managers through Monte Carlo risk analysis. Management has a direct measure of the approximate risk and the chances of achieving various rates of return.

Analysis of Results and Conclusions

Table 1 shows the calculated statistical measures for the six rate of return distributions obtained from the computer simulation.

Statistical Measures

A comparison of the means and standard deviations supports the observations concerning the frequency curves for the simulated rates of return. The distributions from the normal, flat, and peaked variability simulations have approximately similar means (151.5, 154.5, 148.1). Their standard deviations bear a relationship corresponding to their assumed type of variability. The standard deviation for the distribution of the simulation using flat variability is larger than the standard deviation for the simulation using normal variability (37.69: 29.04). The standard deviation for the distribution assuming peaked variability is smaller than the normal distribution standard deviation (23.84:29.04). The distributions from the skewed simulations have means (95.4, 195.1) that are considerably different from the means for the normal, peaked, and flat variability simulations. Their standard deviations, however, are similar (30.39, 33.05). The simulation of random variability has a distribution with a

TABLE 1

STATISTICAL MEASURES FOR THE DISTRIBUTIONS
OF SIMULATED RATES OF RETURN

Assumed Type of Variability	Mean	Standard Deviation	Skewness	Kurtosis	Sampling Error
Standard Normal	151.5	29.04	0.2	2.8	0.73
Peaked Normal	154.5	23.84	0.2	2.7	0.75
Flat Normal	148.1	37.69	0.3	2.7	1.19
Left Skewed	195.1	30.39	-0.1	2.8	0.96
Right Skewed	95.4	33.05	0.5	3.2	1.05
Random	124.6	50.99	0.2	2.6	1.28

mean (124.6), lower than the means from the three distributions of symmetrical variability but higher than the mean for the distribution of right skewed variability. The standard deviation for the random simulation (50.99) is much larger than the other standard deviations.

Comparing the peakedness and skewness of these distributions to the theoretical normal peakedness (kurtosis value of 3) and to the theoretical normal skewness (skewness value of 0), the distributions for the normal, flat, peaked, and random variability simulations are slightly positive or right-skewed and somewhat more peaked than a theoretical normal distribution. The simulation assuming left-skewed variability produced a distribution that was more left-skewed than the others. In contrast, the simulation assuming right-skewed variability produced a distribution that was more right-skewed and that had more peakedness. The simulation distribution using random variability had the lowest degree of peakedness. The measures of skewness and kurtosis are something of a cross-check on the reliability of the simulation since the calculated measures of skewness and kurtosis indicate approximately normally distributed sampling distributions.

Sampling error is another check on the simulation validity. It indicates the possible error in estimating the actual mean rate of return from the mean of the simulated rates of return because of the random simulated sampling used in the Monte Carlo method. A comparison of the sampling errors and means in Table 1 reveals that the differences between the means for the six simulations are greater than the possible errors which might result from the randomnesss of the simulation.

Analysis of Risk for the Hypothetical Investment

The purpose of Monte Carlo risk analysis is to determine for the proposed investment the variability in possible rates of return and the probability of various returns being achieved. Investment risk is defined as the likelihood of not achieving the expected rate of return and the variability in investment return is generally accepted as a measure of this risk. The frequency of individual rates of return occurring during simulation is a means of constructing an anticipated rate of return distribution. This distribution is the basis for determining rate of return variability and inferring the probability of achieving various rates of return.

The rate of return frequency distributions

TABLE 2

RATE OF RETURN FREQUENCIES AND PROBABILITIES FOR THE SIMULATION ASSUMING NORMAL FACTOR VARIABILITY

Rate of Return Intervals	Frequency 1000 Trials	Frequency 1600 Trials	Percentage 1600 Trials	Cumulative Probability 1600 Trials
0 or less	0	0	.000	1.000
1 to 10	0	0	.000	1.000
11 to 20	0	0	.000	1.000
21 to 30	0	0	.000	1.000
31 to 40	0	0	.000	1.000
41 to 50	0	0	.000	1.000
51 to 60	0	0	.000	1.000
61 to 70	0	0	.000	1.000
71 to 80	2	3	.002	.998
81 to 90	10	16	.010	.988
91 to 100	24	42	.026	.962
101 to 110	41	67	.042	.920
111 to 120	72	113	.071	.849
121 to 130	100	161	.101	.748
131 to 140	119	207	.129	.619
141 to 150	129	216	.135	.484
151 to 160	126	199	.124	.360
161 to 170	130	191	.119	.241
171 to 180	88	142	.089	.152
181 to 190	61	98	.061	.091
191 to 200	42	65	.041	.050
201 to 210	27	37	.023	.027
211 to 220	15	26	.016	.011
221 to 230	9	10	.006	.005
231 to 240	4	6	.004	.001
241 to 250	1	1	.001	.000
251 to 260	0	0	.000	.000
261 to 270	0	0	.000	.000
over 270	0	0	.000	.000

and their means and standard deviations have already been presented for the six simulations of different conditions of factor variability. The mean represents the most likely expected rate of return and the standard deviation represents the variability of investment return or the dispersion of returns about the mean. Considering the risk of the hypothetical investment measured by the mean and standard deviation, the hypothetical investment has generally lower anticipated rates of return with more variability if the factor subjective probability estimates are non-normally distributed than if they are normally distributed. Tables 2, 3, and 4 show the approximate probabilities inferred from the frequency of returns occurring during simulation.

The significant difference between the mean rates of return for the simulations of normal and random variability may be indicative of the weakness of the usual Monte Carlo method of risk

TABLE 3

RATE OF RETURN FREQUENCIES AND PROBABILITIES
FOR THE SIMULATION ASSUMING A RANDOM
SELECTION OF VARIABILITIES

Rate of Return Intervals	Frequency 1600 Trials	Percentage 1600 Trials	Cumulative Probability 1600 Trials
0 or less	5	.003	.997
1 to 10	3	.002	.995
11 to 20	4	.003	.992
21 to 30	15	.009	.983
31 to 40	30	.019	.964
41 to 50	56	.035	.929
51 to 60	60	.037	.892
61 to 70	80	.050	.842
71 to 80	99	.062	.780
81 to 90	108	.067	.713
91 to 100	114	.071	.642
101 to 110	115	.072	.570
111 to 120	116	.073	.497
121 to 130	83	.052	.445
131 to 140	113	.071	.374
141 to 150	111	.069	.305
151 to 160	112	.070	.235
161 to 170	79	.049	.186
171 to 180	62	.039	.147
181 to 190	57	.036	.111
191 to 200	58	.036	.075
201 to 210	44	.026	.048
211 to 220	24	.015	.033
221 to 230	16	.010	.023
231 to 240	15	.009	.014
241 to 250	6	.004	.010
251 to 260	11	.007	.003
261 to 270	1	.001	.002
over 270	3	.002	.000

TABLE 4

RATE OF RETURN FREQUENCIES ASSUMING PEAKED NORMAL, FLAT NORMAL, LEFT SKEWED, AND RIGHT SKEWED VARIABILITY

Rate of Return Intervals	Peaked Normal	Flat Normal	Left Skewed	Right Skewed
0 or less	0	0	0	0
1 to 10	0	0	0	0
11 to 20	0	0	0	3
21 to 30	0	0	0	9
31 to 40	0	0	0	18
41 to 50	0	2	0	37
51 to 60	0	3	0	80
61 to 70	0	4	0	102
71 to 80	0	14	0	113
81 to 90	0	31	0	131
91 to 100	6	52	1	132
101 to 110	21	63	3	97
111 to 120	46	81	2	82
121 to 130	92	101	15	61
131 to 140	145	98	25	37
141 to 150	155	107	34	29
151 to 160	160	114	62	23
161 to 170	140	78	73	25
171 to 180	99	65	105	10
181 to 190	66	44	128	5
191 to 200	33	38	136	4
201 to 210	30	41	120	2
211 to 220	5	30	91	0
221 to 230	2	17	79	0
231 to 240	0	11	60	0
241 to 250	0	6	36	0
251 to 260	0	0	22	0
261 to 270	0	0	6	0
over 270	0	0	2	0
Total Trials	1,000	1,000	1,000	1,000

analysis, since most studies on Monte Carlo analysis of investment risk assume normal variability in their factor estimates. From the probability aspect, the simulation using a random selection of the five types of factor variability should have a rate of return distribution with a mean not significantly different from the normal simulation mean and with a greater standard deviation than the simulation using normal variability. This expectation may be derived intuitively in that a random selection of the five types of basic variability should average out after many selections to a type of distribution that approximates a normal distribution but with a greater variability.

A possible explanation for this significant difference involves the complex effect of probabilistic relationships in the hypothetical investment model. There are eleven variables with two pairs intercorredated. Furthermore, not only are the periods in the rate of return equation intercorrelated, but time itself is a variable. It is possible for the probabilities to combine in an unusual manner not representative of the assumed probability because of the construction of the model.

For example, suppose in the random variability simulation a lower range of factor values of one particular investment factor has a greater influence on the rate of return than the higher range of values. The resulting rate of return distribution would then tend to be right-skewed. The hypothetical model does in fact contain certain factors with distinct ranges such as variable cost or market volume. There is also an upper limit to the quantity of sales and production volume, either of which could truncate the distribution of cash flows. Thus the differences in means could arise because of the model construction. If this is the case and the model is truly representative of typical investment situations, then the effect of non-normal probabilities is important. This effect suggests that the simulation using a random selection of non-normal probabilities is preferable to a simulation using only normal probabilities because it is a more representative measure of risk.

TABLE 5

COMPARISON OF PERCENTAGE DISTRIBUTION BY STANDARD SCORES

Standard Score Interval		Assumed Type of Factor Variability						Theoretical Normal Curve
		Standard Normal	Peaked Normal	Flat Normal	Left Skewed	Right Skewed	Random	
From	To Under							
-3.5	-3.0	.000	.000	.000	.000	.000	.000	.001
-3.0	-2.5	.001	.000	.002	.003	.000	.000	.004
-2.5	-2.0	.015	.013	.008	.022	.010	.008	.017
-2.0	-1.5	.042	.043	.044	.042	.028	.048	.044
-1.5	-1.0	.100	.094	.110	.092	.116	.114	.092
-1.0	-0.5	.168	.176	.169	.138	.177	.173	.150
-0.5	0.0	.190	.203	.195	.216	.204	.178	.191
0.0	0.5	.188	.169	.191	.178	.197	.169	.191
0.5	1.0	.139	.141	.114	.135	.104	.145	.150
1.0	1.5	.083	.080	.080	.098	.078	.089	.092
1.5	2.0	.044	.050	.059	.058	.040	.049	.044
2.0	2.5	.022	.027	.022	.013	.033	.018	.017
2.5	3.0	.007	.002	.006	.002	.008	.009	.004
3.0	3.5	.001	.002	.000	.000	.000	.000	.001
Totals		1.000	1.000	1.000	1.000	1.000	1.000	1.000

Cross-Checking by Standard Scores

An evaluation of the six rate of return distributions by means of standardizing their distributions serves as a cross-check on the validity of the simulation results. Standard score values were calculated for each of the six rate of return distributions and converted to percentage frequency. Table 5 shows these percentage frequencies by standard score and also the percentage frequency for the theoretical standard normal curve. A close similarity exists not only between the six simulations but also between the six distributions and the standard normal curve.

This standard tabulation helps to verify the statistical validity of the computer simulation. It indicates that, after correcting for the different means and standard deviations, there is no significant difference between the six simulations resulting from the simulation method.

REFERENCES

1. Hertz, David B. Risk Analysis in Capital Investment. Capital Investment Series, Harvard Business Review Reprints, January-February, 1964, pp. 95-107.
2. Hess, Sidney W. and Quigley, Harry A. Analysis of Risk in Investments Using Monte Carlo Techniques. Chemical Engineering Progress Symposium Series, No. 42, 1963, pp. 55-63.
3. Hillier, Frederick S. "The Derivation of Probabilistic Information for the Evaluation of Risky Investments." Management Science, IX, No. 3 (1963), pp. 443-457.
4. Sasieni, Maurice; Yaspan, Arthur; and Friedman, Lawrence. Operations Research. New York: John Wiley & Sons, Inc., 1967.
5. Wagle, B. "A Statistical Analysis of Risk in Capital Investment Projects." Operational Research Quarterly, XVIII, No. 1 (March, 1967), pp. 13-32.
6. Weston, J. Fred and Brigham, Eugene F. Managerial Finance. 2nd ed. New York: Holt, Rinehart and Winston, 1966.

A COMPUTERIZED INTERACTIVE FINANCIAL FORECASTING SYSTEM

Philip M. Wolfe
Motorola Inc., Semiconductor Products Division
Donald F. Deutsch
Motorola Inc., Semiconductor Products Division
Phoenix, Arizona

ABSTRACT

A computerized interactive forecasting system has been developed for the Finance Department at the Semiconductor Products Division of Motorola Inc. While being basic in its operation, the financial forecasting system has permitted the analysis of large amounts of data on a timely basis. Interactive profit and loss forecasting models have been developed which represent the various operating levels of responsibility in the Division in both domestic and international areas. The financial analyst, from a remote terminal, can communicate with the computer to access, update and report forecast data as well as manipulate any or all of the financial models. The forecasting system was designed to place complete control of operation in the hands of the users, and also to provide flexibility in adapting to constantly changing financial forecasting requirements.

INTRODUCTION

Prior to 1971, all financial forecasts at the Semiconductor Products Division of Motorola Inc. were generated manually. In general, profit and loss forecasts were prepared every month to project financial activity for the next three months, for the remaining quarters in the year, and for the next year. This was done for some sixty operating segments of responsibility within the Division.

Because these forecasts were prepared manually, many problems occurred:

1. The detail that could be considered in a forecast was limited.
2. Excessive time was spent generating and verifying numbers, thereby prohibiting detailed analysis in preparation of the forecasts.
3. Because of the excessive time spent in preparation of a forecast, errors in the forecast or changes in the assumptions could not be adjusted once the forecast preparation was near completion.
4. No valid procedure for measurement of forecast accuracy existed because of the excessive manual efforts required.
5. No historical data base existed which the financial analysts could readily use to prepare the forecasts.
6. Accountants were being used as clerks to run the calculators in order to prepare the required forecasts.
7. Few "What if" questions were analyzed.

The recognition of these problems resulted in the search for a better method. Since the basic forecasting formats could not be reduced due to the requirements of the Corporate Office, it was decided to computerize the financial forecasting process.

Several companies were visited to ascertain how they accomplished their financial forecasting; financial modeling seminars were attended; and a consulting firm specializing in financial modeling was investigated. Following a very thorough study, the decision was made to develop an in-house system which would facilitate interactive financial forecasting.

A system, called the Profit and Loss (P&L) Forecasting System, was developed through the joint efforts of the Planning Administration, the Financial Analysis, and the Management Science Groups. The aim of this system was to provide a set of interactive timesharing computer models which would depict the P&L structure within the Divisions and which would permit the generation of financial forecasts within that structure.

The specific objectives to be met in developing the P&L Forecasting System were the following:

1. The System must be written in FORTRAN to be run on an in-house GE-430 timesharing computer.
2. The System must provide security.
3. The System must be flexible enough that organizational changes do not require reprogramming.
4. It must be easy to set up and change report formats and then generate reports.
5. It must be simple to enter and retrieve large amounts of data from remote timesharing terminals.
6. The System must be general enough to facilitate other modeling efforts.
7. The System must be simple enough to use such that a programmer is not required to run the system.
8. The development and first implementation must not cost more than $25,000.
9. The System must be developed and implemented within 3 months elapsed time (7 man-months).

These objectives which were fulfilled resulted in a general Modeling System which has facilitated the development of other financial and also nonfinancial models at this Division.

P&L FORECASTING METHODOLOGY

Each month, a P&L forecast is prepared for the Semiconductor Products Division. This forecast is divided into two major segments, Domestic and International, and is broken out for the next three months, for the remaining quarters in the year, and for the next year. Also contained in this forecast are historical results for the last month and year-to-date. This forecast is prepared at five levels of management under the Domestic and International segments (see Figure 1). Forecasts are prepared for approximately sixty operating segments of responsibility within the Division. Each of the reports contains approximately seventy line items covering sales, direct and indirect costs, and profit categories.

This monthly P&L forecast is a principal operating document for the Division. It provides goals and measures of performance for Division and Corporate

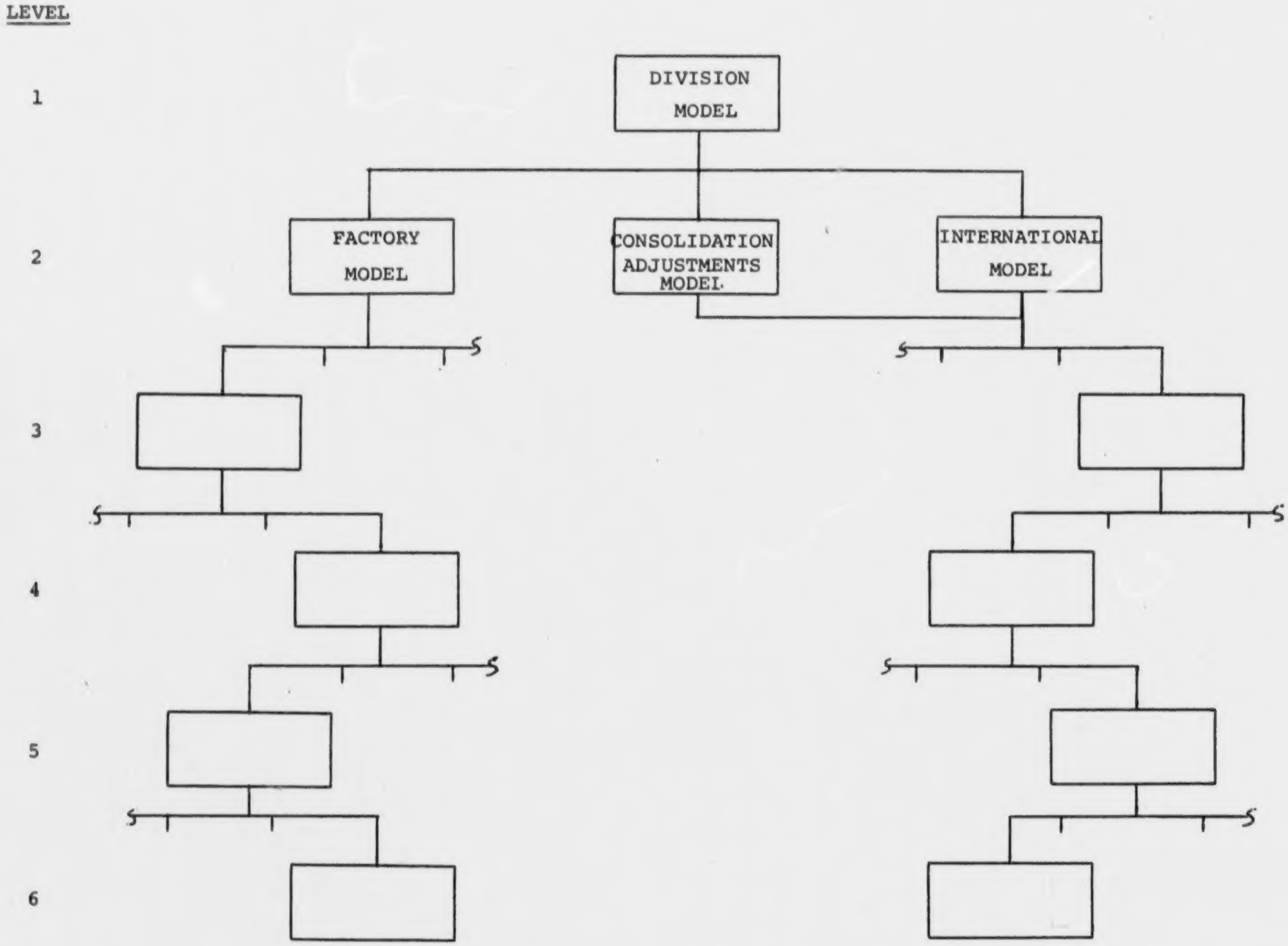

LEVEL
1
2
3
4
5
6
DIVISION MODEL
FACTORY MODEL
CONSOLIDATION ADJUSTMENTS MODEL
INTERNATIONAL MODEL

FIGURE 1

P&L FORECASTING SYSTEM MODEL STRUCTURE

management. It is also used for budgetary control.

The preparation of the P&L forecast is complicated by involved consolidation adjustments between the Domestic and International segments. Also, the Domestic P&L's represent complete manufacturing and sales entities while the International P&L's differ in that some represent such things as sales subsidiaries and assembly plants. In addition, actual results and forecasts are received from around the world that must be analyzed and validated as being reasonable. Another complicating factor is that, when a line item is forecast for as many as nine time periods (months, quarters, years), the coefficients in the function relating that line item to another line item may be different for all nine time periods. For some time periods, there may be no relationship at all; instead the analyst must input the forecasted value. The indirect costs provide additional problems in that suitable bases must be maintained to allocate these costs, and these costs must be forecasted and allocated to the various operating segments of the Division. Thus, because of the size and complexity of the monthly forecast and the fact that it was prepared manually using calculators, little simulation of alternate courses of action could be performed.

Furthermore, before the P&L forecast was modeled, many human errors were made which if detected too late, could not be corrected, or which would remain undetected. Also, the forecast required many weeks to be prepared; consequently, many of the figures were invalid by the time the Division forecast was released.

Once the P&L Forecast was modeled, several things have changed concerning the way a forecast is made and the amount of "what if" simulation that can be performed. The financial analyst has more time to prepare the forecast; therefore, more time can be spent in analysis. Also, a historical data base now exists to aid in the analysis and the measurement of forecast accuracy. More detail (more line items and time periods) is now included in the forecast; consequently, management has more visibility. "What if" simulations may now be run at any level modeled within the Division and the impact evaluated throughout the Division.

THE P&L FORECASTING SYSTEM

Basically, the P&L Forecasting System is composed of a set of FORTRAN programs which are called the Modeling System, and another set of FORTRAN programs called the P&L Model Logic which are linked to the Modeling System to perform the P&L calculations. The Modeling System is essentially independent of the P&L Forecasting Application; therefore, it can be utilized for other applications.

The Modeling System may be defined as an integrated set of computer programs which provide a user with capabilities for creating a structure of models and associated data files, for identifying and organizing the data within the data files; and for accessing, storing, and displaying the data from the data files. The functions of the Modeling System are very general. The P&L Forecasting System utilizes the Modeling System in conjunction with the P&L Model Logic developed for preparing financial forecasts. A general description of this System is given in Figure 2.

In the P&L Forecasting System, a model is a set of mathematical equations which represent the sales revenues and production costs of a segment of the Semiconductor Products Division such as Domestic Factory. The mathematical equations in a model define line items such as Direct Materials cost. Thus, given an expected sales level for a time period, each model will compute a profit forecast for the segment of the Division that it represents.

The models in the P&L Forecasting System have a specific arrangement, a model structure, defined according to their functional interrelationships. These interrelationships are defined by the management organization and by financial reporting procedures. The typical model structure is depicted in Figure 1. The location of a specific model in the model structure is controlled with a code (model code) associated with each model. Thus, the model structure is flexible and can be easily changed in the management organization and/or the financial reporting procedures should change.

Each model in the P&L Forecasting System has associated with it a model data file which is a location in the computer for storing an array of information. The array consists of rows which contain line items and columns which represent elements (time periods). A model data file organization is application dependent. Figure 3 depicts the typical organization of a model data file used in the P&L Forecasting System application.

The first 26 elements (1-26) of a data file comprise space allocated to history. Elements 27-44 are allocated to forecasts, and elements 45-60 are allocated to parameters used to calculate the forecasts. The parameter used to calculate a particular forecast is stored in the same row and 17 elements to the

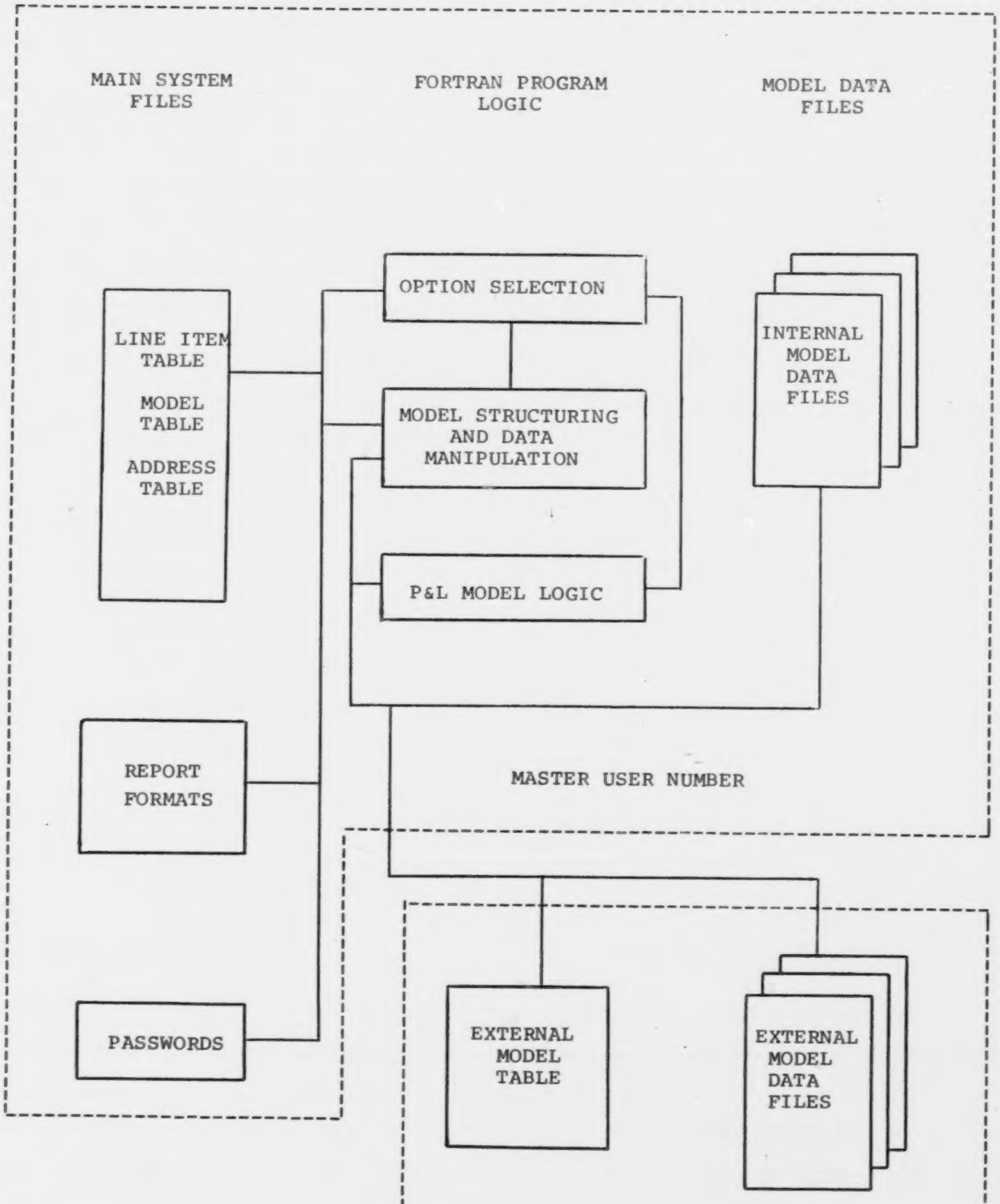
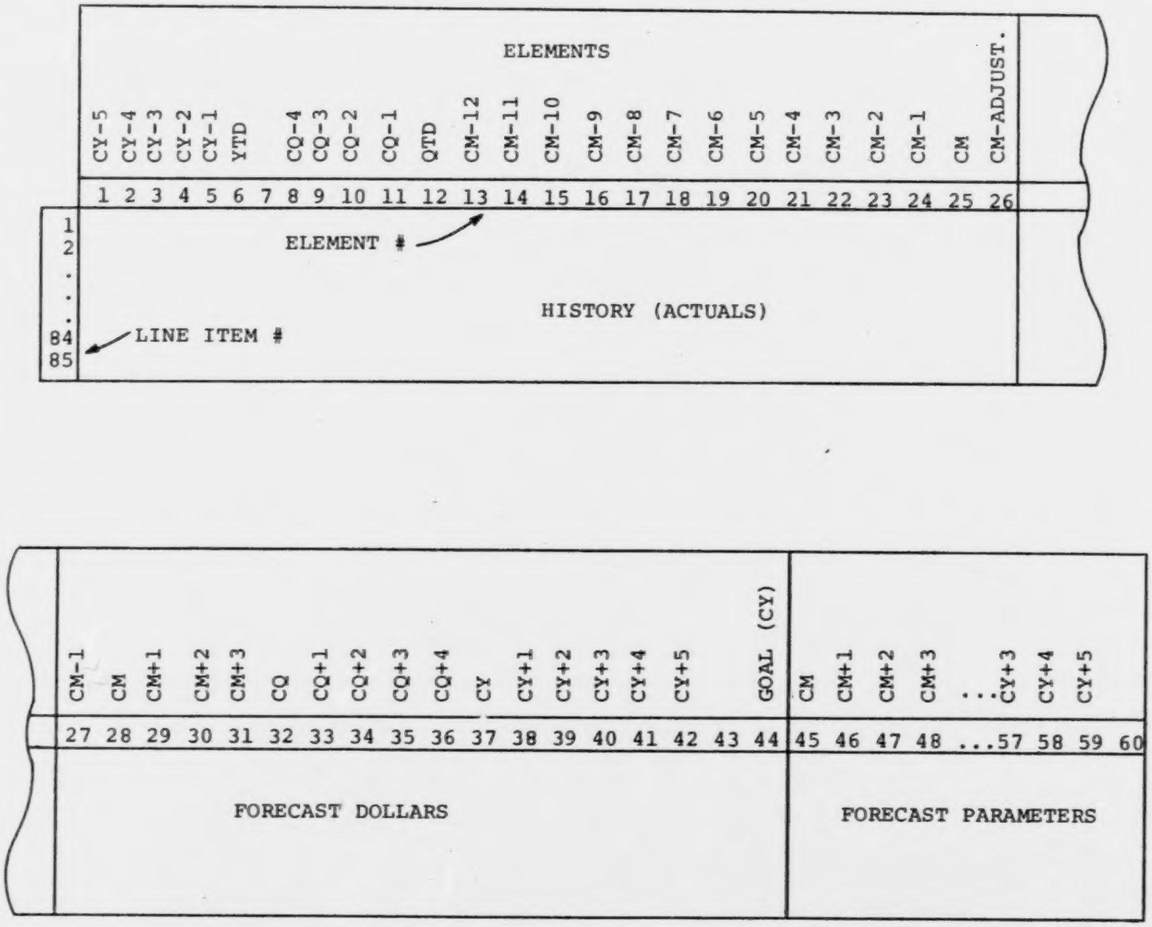

MAIN SYSTEM FILES
FORTRAN PROGRAM LOGIC
MODEL DATA FILES
OPTION SELECTION
LINE ITEM TABLE
MODEL TABLE
ADDRESS TABLE
MODEL STRUCTURING AND DATA MANIPULATION
INTERNAL MODEL DATA FILES
P&L MODEL LOGIC
REPORT FORMATS
MASTER USER NUMBER
PASSWORDS
EXTERNAL MODEL TABLE
EXTERNAL MODEL DATA FILES
ANY USER NUMBER

FIGURE 2

P&L FORECASTING SYSTEM CONFIGURATION


ELEMENTS
CY-5
CY-4
CY-3
CY-2
CY-1
YTD
CQ-4
CQ-3
CQ-2
CQ-1
QTD
CM-12
CM-11
CM-10
CM-9
CM-8
CM-7
CM-6
CM-5
CM-4
CM-3
CM-2
CM-1
CM
CM-ADJUST.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
1
2
.
.
.
84
85
ELEMENT #
HISTORY (ACTUALS)
LINE ITEM #
CM-1
CM
CM+1
CM+2
CM+3
CQ
CQ+1
CQ+2
CQ+3
CQ+4
CY
CY+1
CY+2
CY+3
CY+4
CY+5
GOAL (CY)
CM
CM+1
CM+2
CM+3
...
CY+3
CY+4
CY+5
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48 ...57 58 59 60
FORECAST DOLLARS
FORECAST PARAMETERS

CY = CURRENT YEAR

YTD = YEAR TO DATE

QTD = QUARTER TO DATE

CQ = CURRENT QUARTER

CM = CURRENT MONTH

FIGURE 3

MODEL DATA FILE LAYOUT

right. Thus, if element 28 line item 15 contains the forecast for Direct Materials for the current month, then the parameter used to calculate the forecast is stored in element 45 line item 15. Some exceptions are elements 27 and 44 (CM-1 and Goal elements) for which no space is allocated for parameters.

The P&L Forecasting System is maintained under a master user number and is available to users under any other user number provided the proper access privileges have been given. Under the master user number are stored the main system files, the P&L Forecasting System programs, and the internal model data files which contain the master or permanent P&L information.

The Main System files contain information which identifies the line items, the models, and the report formats to be utilized in the operation of the Forecasting System, and also the passwords which provide access to the System for various users. These files are created through the Forecasting System with the exception of the password file which must be generated outside of the System.

A user from any other user number, if given a Modeling System password, may access the P&L Forecasting System. This access allows a user to perform many of the System functions under his own user number. Such a user may duplicate certain of the internal model data files under his user number and then operate on them independently of the internal model data files. These duplicate files are called the external model data files. It is also possible to transfer information back and forth between the internal and external model data files.

WHAT CAN BE DONE WITH THE FORECASTING SYSTEM

Modeling System Major Functions

The major functions of the Modeling System are described below:

1. Model Structuring
2. Data Storage and Retrieval
3. Model Execution
4. Report Generation
5. Sensitivity Analysis
6. External Model Capabilities
7. Security Procedures

Model Structuring

This function entails creating the main system files and initializing their contents as well as providing the user with the facility of building and maintaining a structure of models. The creation of the main system files provides space for storing such things as line item and model descriptions, data file names, record addresses, and model structure codes. This function also provides the facility for defining and changing line item and model descriptions, creating and deleting model data files, and listing existing line items and model data files.

Data Storage and Retrieval

The inputting, retrieving, and listing of data from the model data files is handled by this function. The data may be input through a terminal or from an existing file. Likewise, data may be retrieved from model data files to be listed at the terminal or placed in a file and listed later at the terminal or at an off-line high-speed printer.

Model Execution

The execution of a model involves executing the model logic (mathematical equations) that make up that model. Values are calculated for a specific time period for the appropriate line items.

In the P&L Forecasting application, some of the functions performed with model execution are:

1. Calculate a P&L for the lowest level model (level 6 in Figure 1).
2. Execute a P&L model above level 6 by executing the associated level 6 models, and calculate parameters for those models above level 6.
3. Calculate a P&L for a higher level model (levels 1-5) without first executing models at level 6.
4. Calculate consolidation adjustments for the Division.

Model execution is a function of the particular application.

Report Generation

The Modeling System permits the analyst to set up report formats interactively through a remote terminal and to write these reports either to a terminal or to a file. If written to a file, the reports can be printed later at a terminal or at an off-line high-speed printer. The report formats may be general with the user specifying what information is to be printed at the time a report is written, or the report can be so specific that the user only specifies what report is to be printed. Once a report format is created, it is saved and may be used again.

Sensitivity Analysis

Management may see the impact of changing the values of variables in the models. This may be done by changing a value in a model data file, then executing that particular model, and then by either listing out values for line items of interest either through a data file list option or through a report print option. Another way is through a sensitivity analysis option which permits the analyst to specify a range of values for a particular line item that is to be changed and the increments of change. The analyst must also specify the line items of interest for which he would like to see values once the desired model is executed. The sensitivity analysis option does not disturb the contents of the data file associated with the model on which the analysis is performed.

External Model Capabilities

The Modeling System provides the capabilities of creating duplicates of the internal model data files under other user numbers. Data may be moved back and forth from internal and external files as well as between external files. Although anyone with the proper password clearance can move data from an internal file to an external file, only one password is cleared to move data into the internal files from external files. This controls the integrity of the internal data files.

Once external model data files exist, the analyst can perform desired "what if" analysis using these files. This permits extensive analysis with the capabilities of restoring a model data file to the initial status simply by recopying desired data from the internal model data files. Essentially, any analysis and report printing performed with the internal model data files can be performed using the external model capabilities of the Modeling System.

Security Procedures

A user must have a Modeling System password assigned to him. This password gives him access to specific models and options of the Modeling System. Thus, a user may have access to only one model and may only list information from the associated model data file using an external list option (see Modeling System Options), while another user may have access to all models and may manipulate the associated data and generate reports as he desires, using any of the external and internal options.

Modeling System Options

The major functions of the Modeling System are performed by the user specifying an option number when prompted by the System. There are twenty explicit options:

1. Initialize System Files.
2. Update Line Item Table. Permits adding, deleting and changing of line item descriptions.
3. Update Model Table. Permits adding, changing and deleting

of model data files, model data file names, and model descriptions.

4. Construct Report Formats. Permits definition of complete or partial report formats. Input may be via a terminal or a file.
5. List Model Table. Permits listing of a table containing names of all existing model data files and the associated model descriptions.
6. Update Model Data. Permits modifying data in any model data file, one or many values at a time. Also permits duplicating all or portions of one data file in another data file as well as resetting all or portions of any data file to zero. Input may be from a terminal or from a file.
7. Execute Models. Permits execution of model logic, which is dependent upon the application.
8. List Line Item Table. Permits listing of any or all existing line item descriptions.
9. List Valid Models. Permits listing of any or all models validated for the current user.
10. List Model Data. Permits listing of any data in a model data file; one or many values may be listed at one time. Output may be via a terminal or a file.
11. Print Reports. Permits the printing of reports whose formats have been defined through Option 4. Output may be via terminal or file with the capability of printing the file using an off-line high-speed printer.
12. Update External Model Table and Move Data Between Files. Permits the creation and deletion of external model files, the moving of data between internal and external files, and the moving of data among external files.
13. List External Model Table. Performs the same function for external models as Option 5 does for internal models.
14. Update External Model Data. Performs same function as Option 6.
15. List External Model Data. Performs same function as Option 10.
16. Execute Models From External Data Files. Performs same function as Option 7.

17. Perform Auto-Sensitivity Analysis. Permits varying any line item for any model over a range of values, executing the proper models, and then listing any desired line items after each model execution.
18. Print Reports From External Data Files. Performs same function as Option 11.
19. Update And Age Model Data. Updates the System time and ages data when the System time changes, i.e., shifts data to the proper elements within the model data file to reflect the correct relative time for that data.
20. Update And Age Model Data In External Files. Performs same function as Option 19.

Example: "What If" Simulation

Assume that for the Model 1, Silicon Transistors, it is desired to vary the Net Sales (line item 10), for the next month (element 28), between the values of $1,000 and $2,000 in steps of $500, and to recompute the complete P&L Statement for each sales value. However, because of the time required to print all the line items, only those specified will be printed, which in this case are line items 10, 15-20, 30, 50 and 82. Along with the forecast dollars, each line item value is computed as a percentage of Net Sales and is listed to the right of the forecast value. It is possible to perform this analysis using Option 17, automatic sensitivity analysis. The analysis is shown in Figure 4.

SYSTEM ACCEPTANCE AND ENHANCEMENTS

The initial P&L Forecasting application was not very sophisticated from a simulation viewpoint, although it did permit extensive "what if" sensitivity analysis. It performed the specific function for which it was developed. In addition, it brought more sophistication to the financial analyst and more accurate and timely reports to management. The acceptance by the users (financial analysts) was greater than expected as the first year billings (storage, connect time, and cpu time) were approximately $75,000 at the GE Mark II billings rate, but much less at the in-house rate.

A significant by-product of this effort was a sophisticated general Modeling System with the following special features.

1. Timesharing random access of data.
2. External model capabilities.
3. High-speed printer output.

```
** OPTION ** ?17

ENTER MOD #; ELMNT #; L.I. # FOR ANALYSIS  ?1;28;10

ENTER START, END VALUES; INC; L.I. #S TO LIST
?1000,2000;500;10,15-20,30,50,82

MODEL   1: SILICON TRANSISTORS

ANALYSIS ELMNT 28, L.I. 10, VALUE          1000
FORECAST ELMNT 28

NET SALES                1000   100.0
DIRECT MATERIAL           110    11.0
FRT,SCRP,SHR,&OBS           6     0.6
DUTY                        1     0.1
DIRECT LABOR               80     8.0
DIRECT LABOR BURDN        150    15.0
DIRECT LABOR FRING          3     0.3
COMMITTED COSTS            21     2.1
TOTAL INDIREC COST        124    12.4
NET PROFIT (LOSS)         252    25.2

ANALYSIS ELMNT 28, L.I. 10, VALUE          1500
FORECAST ELMNT 28

NET SALES                1500   100.0
DIRECT MATERIAL           165    11.0
FRT,SCRP,SHR,&OBS           9     0.6
DUTY                        2     0.1
DIRECT LABOR              120     8.0
DIRECT LABOR BURDN        225    15.0
DIRECT LABOR FRING          4     0.3
COMMITTED COSTS            21     1.4
TOTAL INDIREC COST        124     8.3
NET PROFIT (LOSS)         415    27.7

ANALYSIS ELMNT 28, L.I. 10, VALUE          2000
FORECAST ELMNT 28

NET SALES                2000   100.0
DIRECT MATERIAL           220    11.0
FRT,SCRP,SHR,&OBS          12     0.6
DUTY                        2     0.1
DIRECT LABOR              160     8.0
DIRECT LABOR BURDN        300    15.0
DIRECT LABOR FRING          5     0.2
COMMITTED COSTS            21     1.0
TOTAL INDIREC COST        124     6.2
NET PROFIT (LOSS)         578    28.9
```

FIGURE 4

AUTOMATIC SENSITIVITY ANALYSIS

4. User-controlled report generator.
5. Selective security procedures.
6. General sensitivity analysis.

The Modeling System has facilitated the development of other financial and non-financial models. This system has been used to develop the following interactive models:

1. Five Year Sales Forecast.
2. Production Control Sales Forecast & Billing Report.
3. Marketing Sales Forecast.
4. Corporate Consolidated Balance Sheet.
5. Budget System.

In addition, a batch version has been developed to run on IBM 360 equipment that is being used in Europe to prepare P&L Forecasts for the European subsidiaries. A batch version was necessary as the Corporation has no timesharing facilities in Europe.

Enhancements are periodically being made to the Modeling System. Statistical routines are being added to aid in forecasting, the report generation and writing has been made more general, and data storage has been made more efficient. In addition, a version is being developed to run on IBM's Timesharing Option (TSO).

MULTINATIONAL CAPITAL BUDGETING: A SIMULATION MODEL

André Fourcans

Indiana University

Thomas J. Hindelang*

Indiana University

Abstract

The rapid growth of multinational corporations has hastened the need for the development of robust models to handle the increased risk and complexity. Particularly in capital budgeting, careful analysis and adequate reflection of the critical variables are essential. The great number of relevant variables, their significant interrelationships, and the high degree of uncertainty render mathematical models highly complex or infeasible to solve.

To overcome these shortcomings, a "Hertz-type" simulation model is formulated for the multinational firm. The important international variables--foreign exchange rates, foreign tax methodology, host government controls, and other social, economic, and political factors--are reflected in the model. A two stage approach is utilized: first, investment projects are analyzed by the subsidiary and if they pass this first screening they are proposed for the parent's consideration; second, the parent evaluates the attractiveness of projects from its point of view and ranks proposals for acceptance considering all global opportunities. The model is designed so that sensitivity analysis can be easily performed.

---

*The authors gratefully acknowledge the helpful comments of Dr. W.C. Perkins, Dr. John F. Muth, Dr. Donald Tuttle, Dr. Larry Merville, and the members of our doctoral seminar in international finance at Indiana University.

I. INTRODUCTION

Unquestionably, the capital budgeting process is of critical importance in helping firms achieve their various objectives. Hence, the more accurately capital budgeting models reflect the actual conditions faced by firms, the greater will be the assistance the models provide in reaching corporate goals. The past decade has seen a significant evolution of the methods utilized in capital budgeting. Payback and return on investment approaches have been recognized as inferior to the discounted cash flow models--net present value and internal rate of return. In addition, the importance of reflecting "risk" has become accepted because of the inadequacy of a single parameter (the expected value or mean) to incorporate all of the relevant underlying aspects of cash flow distributions. Therefore, measures of variation or dispersion are called upon to enrich the formulations. However, such extensions are not free of theoretical and applied difficulties. To further complicate matters, correlations among investment opportunities, multiperiod capital rationing, reinvestment of cash throwoffs, and financing decisions should also be reflected. An even more challenging task is at hand when the dimensionality of the problem is still further increased by considering the multinational setting. This necessitates the incorporation of foreign exchange rates, differential foreign tax treatment, and international economic, social, and political factors. Unfortunately, traditional approaches, as well as mathematical programming formulations, have been found wanting in their ability to precisely and robustly reflect the multidimensional setting while rendering a model which can be accurately solved within "reasonable" computer time and memory requirements. These shortcomings make it advantageous to consider Monte Carlo simulation as a natural resolution of the dilemma. Simulation can effectively be used to reflect diverse and complex interrelationships among stochastic variables over a series of years. Key decision variables can be ascertained. The sensitivity of results to changes in state and decision variables can also be determined. Various assumptions relative to the shape and parameter values of input variable distribution can be tested and their impact pinpointed. Hence, simulation proves to be a flexible and powerful approach to the multinational capital budgeting process.

This paper extends the Hertz (see Hertz [16] and [17]) simulation model to the multinational capital budgeting process. The critical areas discussed above are incorporated. The approach is flexible enough to utilize several criteria in the final investment selection process.

The next section surveys the traditional, mathematical programming, and simulation-based capital budgeting models developed to date. Section three investigates the important international variables and their estimation. It also includes an analysis of the parameters, exogenous and endogenous variables, and identities of the model. Section four presents the model's output and supplementary analysis effective in selecting profitable capital investments. Lastly, extensions of the model are discussed.

II. CAPITAL BUDGETING MODELS

To appreciate the great value of simulation techniques, a survey of the predominant capital budgeting models is necessary. For interested readers, various references are given. This part of the paper investigates briefly three groups of capital budgeting techniques: (1) traditional models; (2) mathematical programming models; and (3) simulation models.

Traditional Models

Model builders distinguish three areas of consideration concerning knowledge of the future: (1) "certainty"--where perfect knowledge is assumed; (2) "risk"--where only the parameters and shapes of the probability distributions of future occurrences can be specified; and (3) "uncertainty"--where neither all possible states of the world nor the probability of their occurrences can be specified. Due to the wider knowledge of the "certainty" techniques and the overwhelming difficulty of the "uncertainty" assumptions, the "risk" case will receive our greatest attention.

The traditional "certainty" capital budgeting models (payback, return on investment, net present value - NPV, and internal rate of return - IRR) only require a very brief comment. More and more firms are discarding the former two techniques in favor of one of the latter two. The major reasons for this switch to the NPV and IRR models are that: (1) they accurately reflect the time value of money (a dollar of cash inflow today is more beneficial than a dollar a year from now) which is ignored by the former two; and (2) they consider the importance of the financing decision relative to the investment under consideration. Thus, any sophisticated approach to the capital budgeting process should incorporate these two important aspects. For a discussion of these models see Johnson [19], Bierman and Smidt [1], or conventional finance texts as [31] and [33].

The "risk" case is generally treated through one of the following three approaches: the informal reflection of risk, the risk-adjusted discount rate, and the certainty equivalent. These three methods reflect the fact that future events are unknown but probability distributions can be used to specify the likelihood of various occurrences. The mean and standard deviation of these distributions are used to provide decision criteria.

The informal method subjectively evaluates the tradeoff between the riskiness of projects and their net present values. If two projects are similar in terms of their riskiness (i.e., the standard deviations of the discounted return distributions are approximately equal) the one with the higher mean NPV would be selected. Weston [33, Ch. 8] gives a good illustration of the application of this technique.

The risk adjusted rate of return model

classifies investment proposals according to their riskiness based on the standard deviation of the cash inflows over the life of the investment. Then, the cash inflows are discounted at a rate dependent upon the risk class that the proposal falls into--the riskier the project the higher the rate. The magnitude of the risk adjustment should reflect both the riskiness of the project per se and the firm's attitude toward risk taking.

In the certainty equivalent method, "risky" future cash flows are weighted by a coefficient reflecting the investment's degree of risk (the greater the risk the lower the coefficient). These figures are then discounted at a "risk free" rate (e.g., the rate of interest on U.S. Governmental Bonds). References discussing these latter two models would include Van Horne [31], Weston [33], and Robichek and Myers [27].

These three models are a step in the right direction in that they attempt to reflect the stochastic nature of future events. Needless to say, their major shortcoming is their simplicity --they fail to reflect many very relevant and vitally important aspects: interrelationships among investment opportunities and current operations, various contingencies over the life of the investment, and the capital rationing phenomenon. In addition, the multinational dimension necessitates the incorporation of new relevant variables. To overcome these shortcomings, decidedly more robust mathematical programming models were necessitated. Such models were formulated; their major characteristics are now discussed.

Mathematical Programming Models

Four classes of mathematical programming models have been applied to the capital budgeting process under risk: quadratic programming, dynamic programming, stochastic linear programming, and chance constrained programming. Only a brief and general description of these approaches, and some of the major contributions made, will be undertaken.

Quadratic programming is a technique used to optimize a non-linear objective function subject to linear constraints. One of the pioneering works in this area was by Farrer [10]. He reflected uncertainty in the objective function by using both the mean and variance of the net present value distributions of the investment proposals under consideration. This technique also enables the incorporation of project interdependences through the use of the variance-covariance matrix. Cohen and Elton [6] used this approach in the QP model they formulated. They generated an "efficient set" of achievable risk-return tradeoffs given the feasible combinations of investment alternatives. Unfortunately, as the number of projects grows, difficulties in the solution of the problem escalate quickly.

From a conceptual point of view, a very powerful way of handling uncertainty in mathematical programming models is dynamic programming. This technique optimizes a recursive functional describing a sequential, multi-stage decision-making process where some of the variables are stochastic. Unfortunately, as the number of variables and/or the number of constraints on the problem increase (as is certainly the case with even a small real world problem) the "curse of dimensionality" prohibits efficient solution of the model within realistic computer time and memory requirements. Some advances have been made by Weingartner and Ness [26], Glover [13], and Nemhauser [25] to improve the strengths of computer solution algorithms.

Stochastic linear programming is an approach used to solve problems where the parameters of the model are uncertain but their distributions can be specified. The method involves generating an empirical distribution for the optimum value of the objective function. This is done by allowing the parameters of the system to vary according to their probability distributions and resolving the problem. Cohen and Elton [6] and Byrne, Charnes, Cooper and Kortanek [2] have applied this technique to the capital budgeting area.

Chance constrained programming optimizes an objective function with stochastic variables, and constraints which are only required to hold with some probability less than unity. In this area, four major uninational capital budgeting contributions have been made: Nasland [22], Byrne [3], [4], and Hillier [18]. In addition, Merville [21] has formulated a chance-constrained programming model for the multinational firm's capital budgeting process. However, the lack of efficient solution algorithms for realistic size problems limits the utility of this approach at the present time.

After this brief review, it is possible to show how simulation techniques can be of superior practicality and applicability. Indeed, all of the mathematical models suffer from one or more of the following serious limitations:

1. The model itself is not sufficiently robust to reflect all of the relevant variables and interrelationships in practical sized problems. This is especially true of stochastic linear programming and chance constrained programming, and to a lesser degree of quadratic programming.
2. The model has conceptual weaknesses which jeopardize the validity of the results obtained. Here again, stochastic linear programming and chance constrained programming are the most faulty.
3. The complexities of the model make the accurate solution of realistic problems difficult at best and infeasible at worst. Dynamic programming under uncertainty and chance constrained programming are weakest in this regard.

Due to these significant shortcomings, simulation approaches offer desirable advantages over mathematical models.

Simulation Models

The pioneering work applying Monte Carlo simulation to capital budgeting was undertaken by Hertz [16] in 1964. His approach considered nine variables: market size, selling prices, market growth rate, share of market, original investment required, residual value, operating costs, fixed costs, and useful life of facilities. The decision maker is asked to provide estimates of the expected values and measures of dispersion for the distributions of each of the nine input variables. The output consists of an empirical distribution of return on investment (ROI). However, this initial work was embedded with three major limitations:

1. Cash flows were not discounted, and hence the timing of flows was not taken into account;
2. No consideration was given to the financing decision for the new investment proposal;
3. Project interrelationships and environmental factors were not taken into account.

In his 1968 article Hertz [17] overcomes the first shortcoming by providing three empirical distributions based on the simulation: the payback criterion, the ROI, and the discounted ROI.

In 1968, Salazar and Sen [28] developed a simulation utilizing Weingartner's basic horizon model and built his constraints for interrelated projects into their formulation. Their simulation reflects two types of uncertainties: (1) environmental uncertainty based on what future economic, social, and competitive conditions may be, and (2) cash flow uncertainties where the mean and standard deviation of the cash flows are considered. The results are analyzed by ranking various portfolios of projects as a function of differing environmental conditions and/or management preferences toward risk and return.

In spite of its increasing importance, the subject of international capital budgeting does not seem to attract financial model builders. This fact is very unfortunate because the rise of the multinational corporation necessitates sophisticated tools of analysis. Simulation is certainly a very adequate technique for handling the complex multinational set-up. It demonstrates the interdependence of variables in the decision process and makes it possible to visualize the dynamics in business decisions. Furthermore, risk (particularly international risk) can be introduced very efficiently into the capital investment activity which can thus be rendered very realistic. As of now, evidently only one simulation formulation has been applied to this field [5]. However, this model does not reflect the crucial multinational variables involved, and consequently is still constructed in an uni-national set up.

The proposed simulation utilizes the strengths of the models developed to date. In addition it reflects the critical international variables and the impact of social, economic, and political factors in the multinational arena of the capital budgeting process.

III. NATURE OF INTERNATIONAL CAPITAL BUDGETING

The somewhat complicated evaluation of investment opportunities in an uninational setting is rendered extremely complex in less familiar environments. Indeed, new financial systems and attitudes, new variables such as exchange rates, tax and interest differentials between countries, joint ventures, etc., necessitate a solid framework of analysis. As mentioned above, the usual mathematical programming techniques of capital budgeting lack the flexibility and generality necessary to handle the complex international problems. Conversely, simulation procedures constitute a powerful approach to incorporate stochastic variables and interrelationships. Simulation is now just coming of age, thus gaining wider acceptance in the business community. Hence the model formulated here should prove beneficial to the management of multinational firms.

The proposed simulation has been made as general as possible while not sacrificing ease of understanding and use. In order to provide adequate information and a flexible analysis, a two stage capital budgeting simulation is recommended. First, the investment is evaluated as a uninational opportunity by the subsidiary proposing it. Then, it must be analyzed from the parent's point of view. This joint evaluation is of paramount importance. Indeed, a plant built in a foreign country can be a very profitable investment in itself, but currency devaluations, tax differentials and/or quantitative controls can make it worthless to the parent company. The proposed model handles both the case where the parent is considering a joint venture as well as a 100% participation. Furthermore, the model allows for simultaneous investigation of several investments.

Because of the unquestioned significance of the international variables in the model's development, a discussion of these inputs and their estimation is now undertaken.

International Variables and Their Estimation

Model generality is obtained by considering all the necessary inputs without breaking them down into their specific components. By so doing, the sophistication of the formulation is enhanced without unbearable complexity in its use.

The following relevant international variables are incorporated:

1. Foreign exchange rate risks
2. Inflation risks
3. Expropriation risks
4. Risks of war
5. Foreign taxation and differential tax treatments between countries
6. Duties, embargos, and quantitative controls

The first four risks are represented by stochastic inputs. The last two international variables can be considered deterministic and known a priori by management. Sensitivity analyses allow considerable testing of the adequacy and relative importance of each input. A range of values and a probability distribution must be specified for each parameter. The following discussion considers the type of information which will enhance the quality of each estimate. Of course, the more accurate the data inputs, the more precise and reliable will be the results.

1. Foreign exchange rate risks:

Changes in foreign exchange rates, and particularly devaluations, can affect considerably a project's worthiness. Without question, the dollar equivalent of profits is decreased when a devaluation occurs in the host country of a subsidiary. Therefore, a careful prediction of foreign exchange rates must be made. Various events contributing to changes in foreign exchange rates must be examined:

a. Direct causes:
   - import surplus crises;
   - government spending abroad;
   - withdrawal of foreign balances;
   - over exporting of long term capital

b. Indirect causes:
   - inflation, particularly relative inflation;
   - political conditions;
   - structural changes within the country;
   - national demoralization;
   - policies of foreign countries on investments

An analysis of these variables should lead to adequate forecasts of foreign exchange rates. For more details the reader is referred to [9] and [34].

2. Inflation risks:

Inflation has a great influence on asset valuation, profits, and credit availability. Consequently, this risk must also be studied by management.

Inflation is often associated with immoderate creation of money. A close look at the changes in the money stock is therefore of crucial importance. Other factors to be examined are government spending and changed restrictions of imported goods (through the balance of payments). A good analysis of inflation in the multinational environment can be found in [14].

3. Expropriation risks:

Obviously, expropriations of the foreign investment are of overriding importance. Therefore, a careful evaluation of the characteristics of nations and their propensity to expropriate must be established. Also, the features of the firms more subject to expropriation must be examined.

a. Country characteristics:
   - GNP per capita: measures the level of development and can be expanded to a ranking of nations according to their propensity to expropriate [15].

- Ideology of the industrial elite: [8] gives a ranking of the elite depending on its willingness to expropriate.
- Public sector-private sector mix: the lower the weight put to the value of private ownership, the higher the propensity to expropriate [29], [30].
- Political stability: a major variable in the assessment of expropriation [11] establishes a rating scale for stability which can be used to determine the risk of expropriation.
- Balance of payments: the worse the balance of payments the more likely it is that the country perceives the repatriation of profits as a threat, which may lead to expropriation.
- Other variables such as the level of the domestic entrepreneurial sector [30], colonial heritage, etc.

b. Firm characteristics:

- Nature of economic activity: from the highest propensity to expropriate to the lowest, investments are classified as follows:
  1. service.
  2. extracting.
  3. public utilities.
  4. agriculture.
  5. manufacturing.
- Importance of the firm in the country's economic system: this factor renders the investment more or less vulnerable.
- Foreign exchange activity of the firm: depending on the contributions to the balance of payments the investment is more or less subject to expropriation.
- Nationality of the firm: for cultural reasons some nationalities are more accepted than others.
- Ownership characteristics: joint ventures are less vulnerable than a branch or a 100% owned subsidiary [12].
- Tactical vulnerability of the firm: product, skills, management style, etc. also influence expropriation.

4. Risk of war:

Whereas expropriation does not necessarily mean complete loss of the value of the assets (because of indemnity from governments), the outbreak of a war can impose a complete loss. Therefore, as elusive as this variable is, an estimate of war possibilities should be made by management. A very useful rating scheme for such an evaluation is made in [8].

5. & 6. Tax treatments, quantitative controls, duties, and embargos:

Tax treatments, quantitative controls, duties, and embargos can be considered as known with certainty even if some changes can occur over the life of the investment. However, the simulation allows probabilistic evaluation of these inputs.

The taxation of funds shifted from a foreign country to the U.S. is a very complex subject, and should be studied carefully for each specific investment. However, it can be stated that a double taxation problem will often occur: funds are taxed by the country or region and possibly taxed again by the IRS. Fortunately, the US has taxation agreements with numerous countries so as to eliminate unfair taxation (for example, profits, even if not repatriated will be taxed at about the same rate as if obtained in the U.S.). Therefore, the taxation of dividends, profits, and royalties and fees is one of the model's inputs both for the subsidiary and the parent.

All the other inputs necessary for the determination of cash flows are straightforward enough and do not create special problems of estimation. Even if the predictions of the international variables discussed above seem somewhat complex, they only require careful evaluation and analysis of available information. Furthermore, such requirements should motivate managers to investigate the international environment and help them understand better the multiple interrelationships inherent to multinational investments. This procedure, in conjunction with the decision maker's judgement, leads to very realistic estimates of the crucial uncertainty profiles.

Project Related Variables and Their Estimation

In addition to the critical international variables, the following inputs are also required in order to ascertain the project's cash flows:

1. Initial Outlay;
2. Financing Costs (for the parent and the subsidiary including principal and interest);
3. Working Capital needs for the project;
4. Market size for product generated by the investment proposal;
5. Market growth rate over the life of the project;
6. Selling prices and demand relationships;
7. Market share achieved by firm;
8. Variable costs per unit;
9. Fixed costs per year;
10. Transportation costs;
11. Useful life and salvage value of the project as well as depreciation method selected;
12. Host country tax on profits generated by the project.

Variables one, two, and twelve have very little uncertainty associated with them; thus, the dispersion in their distributions is small. All of the others are more uncertain and take on distributions of varying shapes and dispersion. Because of widespread knowledge of the meaning and impact of the above variables plus their adequate treatment elsewhere in the literature (see Hertz [16], [17]), further discussion is not felt necessary.

Subsidiary Simulation of the Investment Proposal

As mentioned previously, the nature of multinational capital budgeting decisions necessitates careful evaluation of projects both from the subsidiary's and the parent's point of view. Thus, we will discuss in depth how the simulation proceeds in each of these analyses.

The subsidiary's evaluation of a given investment proposal utilizes mainly the direct project costs and revenues discussed above. The analysis uses a uninational framework and considers the parent mainly as a source of funds to finance accepted projects.

The technical details of this stage of the simulation are presented in three illustrations. Table 1 lists the relevant cash inflows and outflows for the subsidiary.

TABLE 1

SUBSIDIARY CASH FLOWS

Inflows	Outflows
Revenue from Sales	Initial Outlay
Salvage Value	Financing Costs
	Host Country Taxes
	Operating Costs

Table 2 defines the variables (both exogenous and endogenous) and formulates the identities of the simulation model. Figure 1 shows a flow chart of this part of the simulation. Of course, the main results produced by the simulation are the empirical values of net income after host country taxes, and the yearly net cash inflows over the life of the investment. Based on these values, the desirability of the project can be determined using the discounted rate of return, net present value, and the payback criteria. The subsidiary will then either recommend that the project be accepted or rejected based on the empirical distributions of the various criteria mentioned. This decision is made by considering the subsidiary's cost of capital (which can be different from the world-wide cost of capital of the total corporation).

Parent Company's Simulation of the Capital Investment Process

The parent company takes a more global view in its evaluation of potential projects. It utilizes the empirical data relative to the project per se, but also incorporates the critical international variables associated with the transfer of funds. The additional risks and uncertainties discussed above are built into the framework so that the parent can adequately assess the situation before it commits funds to a given project in a specific country.

Table 3 shows the cash flows from the parent's point of view.

TABLE 2

VARIABLES OF THE SUBSIDIARY SIMULATION MODEL

PARAMETERS:

SP_t = Selling price per unit in year t   KS = The subsidiary cost of capital
DR_t = Depreciation rate for year t selected by user
MAX = Total number of simulation runs to be considered

EXOGENOUS VARIABLES:

Stochastic variables with known probability distributions:
MG_t = Market growth rate for each year t
MS_1 = Initial market size in number of units
SM_t = Share of the market for each year t
INV = Initial Investment required by the proposal
N = Useful life of investment
FC_t = Total operating fixed costs in year t
VC_t = Variable Operating Costs per unit in year t
IC_t = Interest costs associated with the project in year t
OC_t = Other project related costs in year t
WC_t = Working Capital Needs of the project in year t
TR_t = Tax rate for host country tax on project returns in year t
IR_t = Rate of inflation in year t
WAR_t = The probability that a war will break out in the host country during year t
$LWAR_t$ = The % of loss suffered by the firm if a war occurs in year t
EX_t = The probability that expropriation will take place in host country in year t
LEX_t = The loss suffered by the firm if expropriation takes place in host country during year t

ENDOGENOUS VARIABLES:

$USAL_t$ = Unit sales generated by the proposal in year t
REV_t = Total revenue generated by the proposal in year t
TC_t = Total costs associated with the project in year t
TAX_t = Host country tax on taxable income generated by project in year t
$NIAT_t$ = Net Income after host country tax generated by project in year t

---

TABLE 2 - CONTINUED

NCI_t = Net Cash inflow generated by project in year t

BV_t = Book value of the project in year t

SV_t = Salvage value of the project in year t

$TINF_n$ = Terminal Inflow if expropriation or war occurs

$PAYB_m$ = Payback period for the investment on the m^{th} simulation run

NPV_m = Net Present value for the investment on the m^{th} simulation run

IRR_m = Discounted rate of return for the investment on the m^{th} simulation run

IDENTITIES:

$BV_0 = INV$

$BV_t = INV - (DR_t)(BV_{t-1})$

$MS_t = (MS_{t-1})(1+MG_{t-1})$ $t=2,3,\ldots,N$

$USAL_t = (MS_t)(SM_t)$ $t=1,2,\ldots,N$

$REV_t = (SP_t)(USAL_t)$ " "

$TVC_t = (VC_t)(USAL_t)$ " "

$DEP_t = (DR_t)(BV_t)$ " "

$TC_t = TVC_t + FC_t + OC_t + DEP_t$ $t=1,2,\ldots,N$

$TAX_t = (TR_t)(REV_t - TC_t)$ " "

$NIAT_t = REV_t - TC_t - TAX_t$ " "

$NCI_t = NIAT_t + DEP_t - WC_t$ " "

$$SV_n = \sum_{t=1}^{n} (INV - DEP_t)(1 + IR_t)$$

If expropriation (EX_n) occurs in year n, determine loss suffered (LEX_n), then

$TINF_n = (1 - LEX_n)(SV_n + NCI_n)$

If war (WAR_n) occurs in year n, determine loss suffered ($LWAR_n$), then

$TINF_n = (1 - LWAR_n)(SV_n + NCI_n)$

$PAYB_m$ = The period i such that: $INV - \sum_{t=1}^{i} (NCI_t + IC_t) = 0$

$$NPV_m = \sum_{t=1}^{n} \frac{NCI_t}{(1+KS)^t} - INV$$

IRR_m = The discount rate r such that: $\sum_{t=1}^{N} \frac{NCI_t}{(1+r)^t} - INV = 0$

Figure 1

FLOWCHART OF SUBSIDIARY SIMULATION

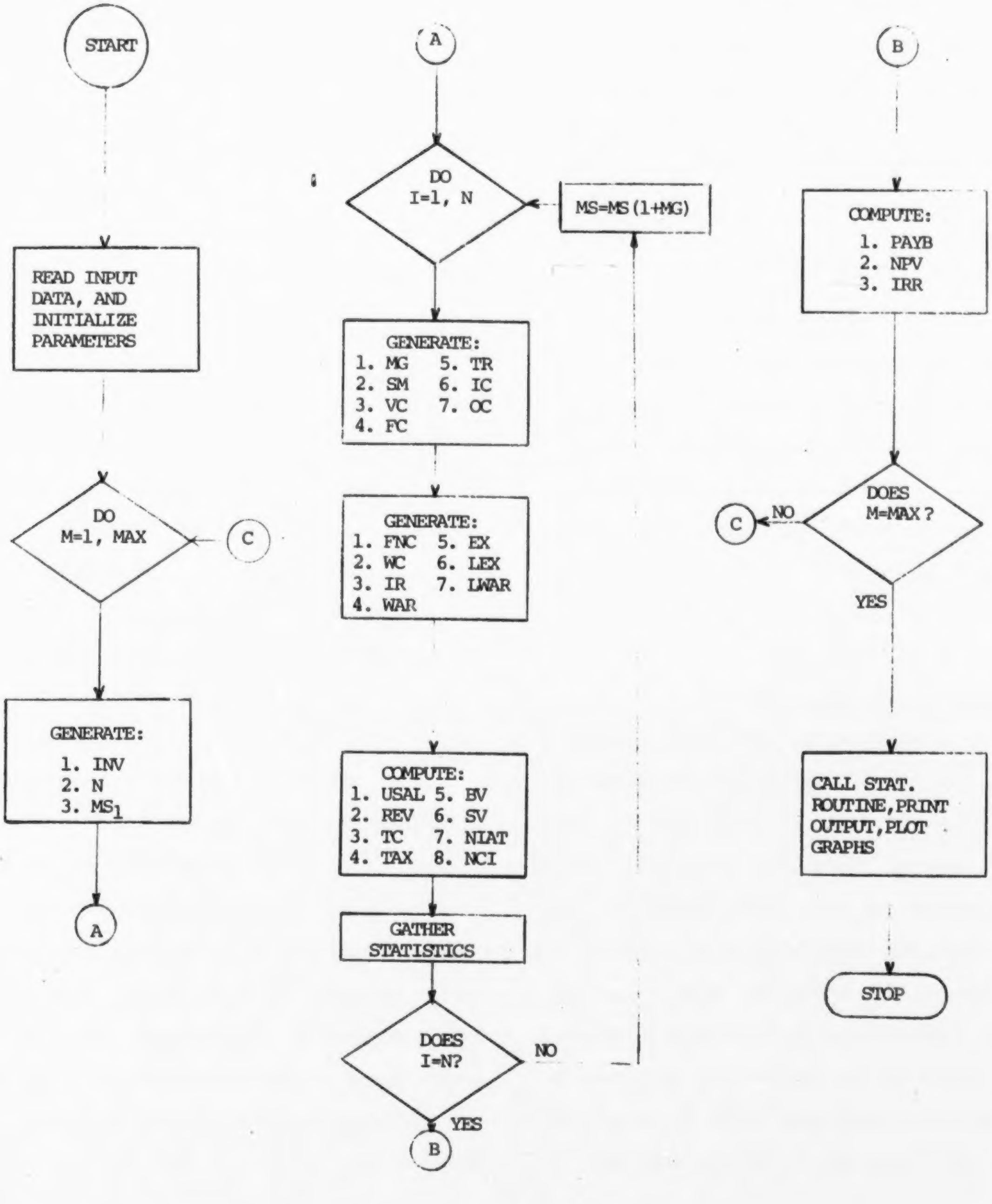

START
READ INPUT DATA, AND INITIALIZE PARAMETERS
DO M=1, MAX
C
GENERATE:
1. INV
2. N
3. MS_1
A
A
DO I=1, N
MS=MS(1+MG)
GENERATE:
1. MG 5. TR
2. SM 6. IC
3. VC 7. OC
4. FC
GENERATE:
1. FNC 5. EX
2. WC 6. LEX
3. IR 7. LWAR
4. WAR
COMPUTE:
1. USAL 5. BV
2. REV 6. SV
3. TC 7. NIAT
4. TAX 8. NCI
GATHER STATISTICS
DOES I=N?
NO
YES
B
B
COMPUTE:
1. PAYB
2. NPV
3. IRR
DOES M=MAX?
NO
C
YES
CALL STAT. ROUTINE, PRINT OUTPUT, PLOT GRAPHS
STOP

TABLE 3

PARENT COMPANY CASH FLOWS

Inflows	Outflows
Direct savings generated by the project	Equity funds provided
Profit repatriated	Loans provided
Dividends	Labor, material, and other costs
Royalties and fees	Transportation costs
Interest and loan repayments	Taxes paid on dividends, royalties, and profits repatriated

Table 4 represents the new variables and identities of importance here. Figure 2 presents the flowchart of the parent's analysis. The same outputs as before--internal rate of return, net present value, and payback--provide the criteria in the parent's evaluation of the worth of the project. The decision is made by using a world-wide cost of capital and any additional qualitative factors.

Mechanics of the Simulation

As noted above, the simulation is designed to be flexible and complete yet not overdemanding on the user relative to necessary data inputs. However, it was also pointed out that the more precise the input specifications are, the more exact and helpful will be the results generated by the simulation. Thus, balancing these tradeoffs, the decision-maker is asked to specify the various variables as accurately as he can for as many years in the future as possible. It is realized, of course, that the farther into the future a user must estimate distributions, the greater the degree of uncertainty. Offsetting this shortcoming are two countermeasures: (1) the discounting process which weights more distant years less heavily, and (2) the fact that sensitivity analysis can be used to determine the impact of changes in the input variables on the decision criteria. In order to make the variable estimation process as painless as possible, the user is given many alternatives as to the method of specifying inputs: (1) he can provide the pessimistic, optimistic, and most likely estimates; (2) the parameters of well known distributions (e.g., Binomial, Uniform, Normal, Beta, etc.) can be specified; (3) he can input any discrete distribution that he feels is appropriate; or (4) he can specify that the distribution is a composite of various distributions. The user is asked to input parameters and distributions for as many years in the future as he feels confident of. However, some variables will incur only minor changes over time, and the distributions can be unchanged for several years.

It is important to describe more precisely how the international aspect of the simulation is handled. The risks of expropriation and war are obtained through a Monte Carlo determination. When the simulation establishes that expropriation or war occurred, it determines, from the input distribution, the associated loss. This result is used to derive the terminal inflow as a proportion of salvage value and the yearly cash inflow.

TABLE 4

VARIABLES OF THE PARENT COMPANY SIMULATION MODEL

PARAMETERS:

DET_0 = The debt funds committed to the project by the parent in year 0

EQY_0 = The equity funds committed to the project by the parent in year 0

DIV_t = The dividend rate as a percent of earnings generated by the project in year t

REP_t = The percent of profits repatriated in year t

KP = The parent company's cost of capital

EXOGENOUS VARIABLES:

Stochastic variables with known probability distributions:

FER_t = The Foreign Exchange Rate in year t

ROY_t = The amount of royalties and fees to be paid to the parent in year t

SAV_t = The direct savings generated by the project in year t

LMC_t = The labor, material, and other costs paid by the parent for production of the product by sub in year t

$TRAN_t$ = The transportation costs associated with importing the product in year t

$PITR_t$ = The weighted "international" tax rate on dividends, royalties and profits repatriated

$PHTR_t$ = Parent home tax rate

INT_t = The interest payments received by the parent in year t

$PRIN_t$ = The principal payments received by the parent in year t

$REQY_t$ = Equity funds retired in year t

ENDOGENOUS VARIABLES:

$PREV_t$ = The before "international" tax total foreign revenue for the parent generated by project in year t

PTC_t = The total cost for the parent generated by project in year t

$PTAX_t$ = The total tax paid by the parent in year t

PITAX = The amount of "international" tax paid by the parent

PHTAX = The amount of home tax paid by the parent

TABLE 4 - CONTINUED

$PNIAT_t$ = The parent's net income after all taxes

$PNCI_t$ = The parent's net cash inflow in year t

$PPAYB_m$ = The parent's payback for simulation run m

$PNPV_m$ = The parent's net present value for simulation run m

$PIRR_m$ = The parent's internal rate of return for simulation run m

IDENTITIES:

$PREV_t = (FER_t)[(DIV_t + REP_t)(NIAT_t) + ROY_t + INT_t]$

$PTC_t = LMC_t + TRAN_t$

$PITAX_t = (PREV_t)(PITR_t)$

$PHTAX_t = (SAV_t - PTC_t)(PHTR_t)$

$PTAX_t = PITAX_t + PHTAX_t$

$PNIAT_t = PREV_t + SAV_t - PTC_t - PTAX_t$

$PNCI_t = PNIAT_t + PRIN_t + REQY_t$

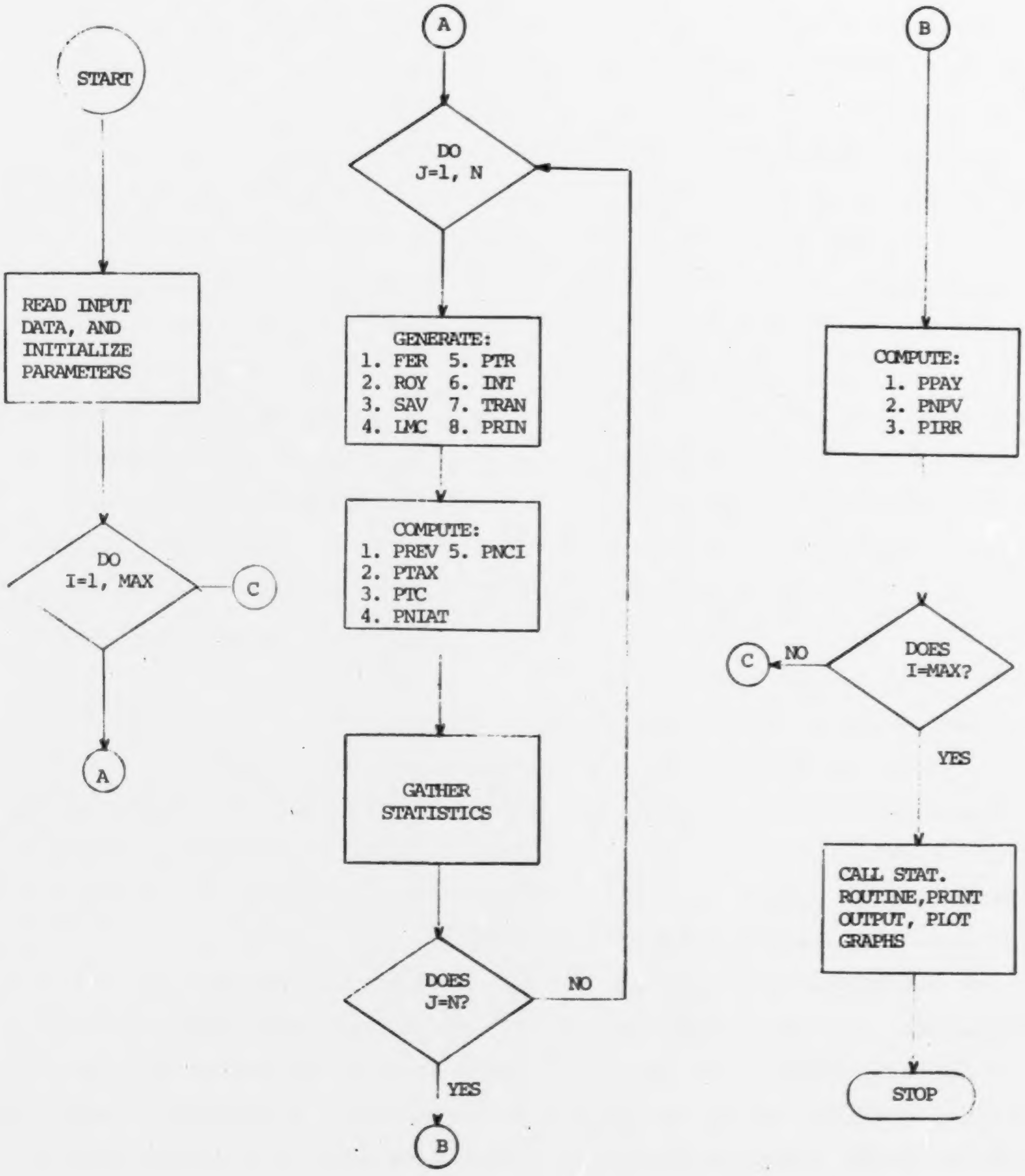
$PPAYB_m$ = The period i such that $(DET_0 + EQY_0) - \sum_{t=0}^{i} PNCI_t = 0$

$$PNPV_m = \sum_{t=0}^{N} \frac{PNCI_t}{(1+KP)^t} - (DET_0 + EQY_0)$$

$PIRR_m$ = The discount rate r such that $\sum_{t=0}^{n} \frac{PNCI_t}{(1+r)^t} - (DET_0 + EQY_0) = 0$

Figure 2

FLOWCHART OF PARENT COMPANY SIMULATION


START
READ INPUT DATA, AND INITIALIZE PARAMETERS
DO I=1, MAX
C
A
A
DO J=1, N
GENERATE:
1. FER 5. PTR
2. ROY 6. INT
3. SAV 7. TRAN
4. LMC 8. PRIN
COMPUTE:
1. PREV 5. PNCI
2. PTAX
3. PTC
4. PNIAT
GATHER STATISTICS
DOES J=N?
NO
YES
B
B
COMPUTE:
1. PPAY
2. PNPV
3. PIRR
C
NO
DOES I=MAX?
YES
CALL STAT. ROUTINE, PRINT OUTPUT, PLOT GRAPHS
STOP

Inflation is dealt with in two ways. First, it can be taken into consideration in the estimation of the exogenous variables by the user's specifying a different distribution for each year of the anticipated useful life of the project. Second, the distribution can be shifted to the right, every year, by the expected percent inflation which can be done for selling price, variable cost, etc. If a single distribution is specified for all periods, the inflation factor is built into the simulation and taken into consideration in the yearly revision of the distributions for the exogenous variables.

It is also important to outline that the model handles dependency among the random variables. Some relationships can be easily taken care of in the estimation of the different distributions. For example, a high rate of inflation in a given year must be associated with larger expected changes in foreign exchange rates for that year, and the corresponding distributions must be so built. However, some dependencies are contingent on the value of the random variables generated by the simulation and can only be handled by the model. An example will make things clearer. It is reasonable to assume that, generally, a high level of fixed cost is associated with a lower variable cost per unit. Consequently, the model takes this fact into consideration and generates low values of variable cost whenever high fixed costs are selected from its distribution. The same type of treatment is established between other interrelated variables.

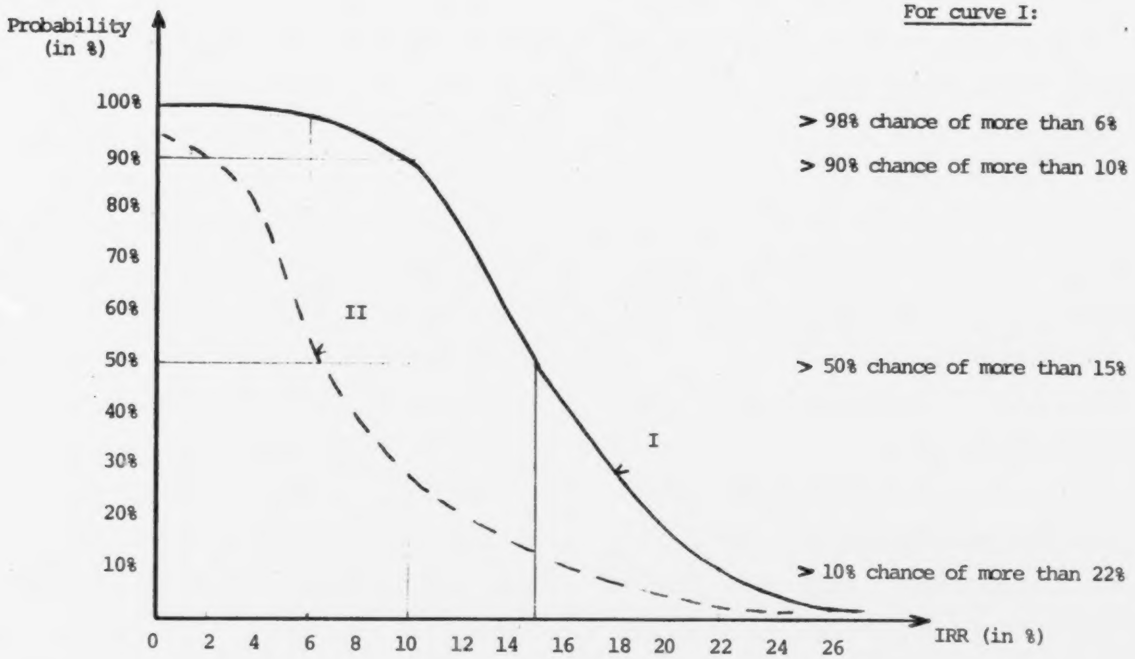
One final detail should be mentioned. Because of the two stage analysis of the investment proposals--first by the subsidiary and then by the parent--two different costs of capital are used. The subsidiary uses its own cost of capital in order to determine whether the investment is desirable from its viewpoint, and if the project should be recommended to the parent for acquisition. In a similar vein, the parent uses a world wide cost of capital figure which it considers relevant (given the risk posture of the investment and the economic, social, and political factors present in the host country) to determine whether it should commit funds to the project. Such an approach gives a double, somewhat independent, more stringent screening of proposals. They must survive both cut off points in order to be adopted by the multinational firm.

IV. VALIDATION AND ANALYSIS OF THE MODEL'S OUTPUT

The simulation not only permits managers to evaluate and compare the performance of different potential investments, but also presents an analytical approach to determine relationships among investment variables and international factors.

The main output consists in the two profiles of Net Present Value (NPV) and Internal Rate of Return (IRR) for the parent company and for the subsidiary. As explained previously, the return to the parent is not the same as to the subsidiary in the country of the investment.

Figure 3



Therefore, a double evaluation of each investment is highly recommended, even in the case of a 100% financing by the parent.

Figure 3 gives an example of the main output. Curve I represents the IRR profile for the subsidiary whereas II is for the parent. As can be quickly noticed in this specific case, the IRR for the parent is everywhere lower than the subsidiary's IRR. However, this need not always be the case (it would depend on the influence of foreign exchange rates, tax differentials, etc.). The purpose of these two profiles is to make sure that the worthiness of the investment can be evaluated by all the groups of the organizations (parent's managers and possible partners in the country of the investment) with their possibly different aspirations. Therefore, an investment is worth having only if these two groups' criteria of acceptability are met.

How are these profiles used? As demonstrated by curve I of Figure 3, there is a 98% chance that an IRR ≥ 6% can be obtained, a 90% chance of more than 10%, a 50% chance of more than 15%, and a 10% chance of an IRR ≥ 22%. We know that the investment will be worthwhile (from the point of view of the subsidiary) if the IRR is at least equal to its cost of capital.

If we assume a subsidiary's cost of capital of 10%, the chance of having an IRR $\geq$ 10% are 90 out of 100. The decision makers will have to decide whether they are ready to take the risk implied: 90 chances out of 100 of having a profitable investment, but 10 out of 100 of losing money. The same analysis needs to be done with curve II from the point of view of the parent (we wish to remind the reader that the cost of capital for the parent and the subsidiary can be different).

The analysis of the output data is rendered more sophisticated than merely evaluating the graphical output by the following elaborations. A statistical analysis subroutine using a multiple ranking criteria discussed by Kleijnen, Naylor, and Seaks [20] analyzes and determines the order of the project desirability and whether statistically significant differences exist among the ranked projects. This analysis is performed by each subsidiary and by the parent for all projects considered by the multinational firm. Such results are invaluable where the firms are faced with capital rationing and multiple, competing opportunities and risks.

In addition, because of the importance of extremes, the simulation could be rerun at least two other times to evaluate the impact of all the inputs having very optimistic distributions and very pessimistic ones. Thus, each investment would have three profiles for each of the criteria. This more complete information can provide valuable insights relative to the investment's overall attractiveness.

Payback criteria are also given as an output of the model through the same type of profiles. This ratio tells the decision maker the number of years required to recover the initial cash investment. This method of investment evaluation should only be used as a secondary criterion, i.e., to differentiate between mutually exclusive projects which have about the same IRR or NPV profiles. However, even if the payback criterion is not a measure of profitability (it does not take into account the cash flows after the payback period) it can be important for international investments. Indeed, the shorter the payback period the smaller the risk of loss due to expropriation, war, or unfavorable foreign exchange rate fluctuations. Therefore, managers can consider this measure as an important aspect of the multinational investment process.

Another significant benefit from the simulation approach is the sensitivity analysis that can be performed. Indeed, decision makers can change the distribution of each variable one at a time, and have a good understanding of the importance each variable has on the value of the investment. It allows an increased comprehension of the relationships among variables and their impact on the decision process. This information is extremely valuable especially for the evaluation of the international variables, particularly for foreign exchange rates which are difficult enough to forecast. If, for example, the final results are found very little affected by changes in currency values, it is clear that the uncertainty of the investment is greatly reduced. On the contrary, high sensitivity to foreign exchange rates would warn the decision maker to give special forecasting attention to this variable.

V. CONCLUSION AND EXTENSIONS

The major emphases of the simulation proposed in this paper were: (1) the extension of capital budgeting analysis to include both project related and international variables relevant to the multinational firm; and (2) the flexibility of a two-stage screening process where first subsidiaries evaluate investment proposals, and then the parent company supplements the analysis by considering the project's desirability from its point of view.

The dual goals of the simulation design were to provide a robust and flexible model and to require only those information inputs that could be relatively accurately estimated. It is because of this second goal that the model does not extensively treat the interrelationships among current proposals and ongoing operations, as well as among the proposals themselves. However, an extension of the current formulation could be made by formally reflecting these portfolio effects. As information systems become more sophisticated, these improvements will certainly become more feasible.

REFERENCES

1. Bierman, H. and S. Smidt, The Capital Budgeting Decision 3rd Edition (N.Y.: Macmillan Co., 1971).

2. Byrne, R., A. Charnes, W. Cooper, K. Kortenek, "A Chance Constrained Programming Approach to Capital Budgeting," Journal of Financial & Quantitative Analysis (Dec., 1967), pp. 339-64.

3. ______, "A Discrete Probability Chance Constrained Capital Budgeting Model I," Opsearch, (Dec., 1969), pp. 171-98.

4. ______, "A Discrete Probability Chance Constrained Capital Budgeting Model II," Opsearch, (Dec., 1969), pp. 226-61.

5. Chambers, J., S. Mullock, and D. Smith, "The Use of Simulation Models at Corning Glass Works," in Corporate Simulation Models, Ed. by A. Schrieber.

6. Cohen, K., and E. Elton, "Inter-Temporal Portfolio Analysis Based on a Simulation of Joint Returns," Management Science, (Sept., 1967), pp. 5-18.

7. Dickson, G.W., J.J. Mauriel, and J.C. Anderson, "Computer Assisted Planning Models: A Functional Analysis," in A.N. Schrieber ed. Corporate Simulation Models, (Seattle, Wash.: College on Simulation & Gaming, 1970), pp. 43-70.

8. Dunlop, Harbism, C. Kerr, and C. Meyers, Industrialism and Industrial Man, (Cambridge: Harvard University Press, 1960).

9. Einzig, P., Foreign Exchange Crises, (N.Y.: Macmillan Co., 1970).

10. Farrer, D.F., The Investment Decision Under Uncertainty, (Englewood Cliffs, N.J.: Prentice-Hall, 1962).

11. Feierbend, I., "Conflict, Crisis, and Collision: A Study of International Stability," Psychology Today, (May, 1968).

12. Friedman, Kalmanoff and Wolfgang, Joint International Business Ventures, (N.Y.: Columbia University Press, 1961).

13. Glover, F., "The Knapsack Problem: Some Relations for an Improved Algorithm," Management Science Research Report No. 38, (1965).

14. Haberler, G., Inflation: Its Causes and Cures, (American Enterprises Institute, Washington, D.C., July, 1966).

15. Harbison, F., and Meyers, C., Education, Manpower, and Economic Growth, (N.Y.: MacGraw Hill, 1966).

16. Hertz, D.B., "Risk Analysis in Capital Investment," Harvard Business Review, (Jan., 1964), pp. 95-106.

17. ______, "Investment Policies That Pay Off," Harvard Business Review, (Jan., 1968).

18. Hillier, F.S., "Derivation of Probabilistic Information for the Evaluation of Risky Investments," Management Science, (April, 1963), pp. 443-457.

19. Johnson, R.W., Capital Budgeting, (Wadsworth Publishing Co., Belmont, California, 1970).

20. Kleijnen, J.P.C., T.H. Naylor, and T.G. Seaks, "The Use of Multiple Ranking Procedures to Analyze Simulations of Management Systems," Management Science (Feb., 1972), pp. 245-57.

21. Merville, L.J., An Investment Decision Model for the Multinational Firm: A Chance-Constrained Programming Approach, Unpublished Ph.D. dissertation, University of Texas at Austin, 1971.

22. Naslund, B., "A Model of Capital Budgeting Under Risk," Journal of Business, (April, 1966), pp. 257-271.

23. Naylor, T.H., Computer Simulation Experiments with Models of Economic Systems, (N.Y.: Wiley & Sons, 1971).

24. ______, ed., The Design of Computer Simulation Experiments (Durham, N.C.: Duke University Press, 1969).

25. Nemhausser, G.L., Introduction to Dynamic Programming, (N.Y.: Wiley & Sons, 66).

26. Ness, D., and H.M. Weingartner, "Methods f the Solution of the Multi-Deimensiona 0/1 Knapsack Problem," Operations Research (Jan.-Feb., 1967).

27. Robichek, A., and S. Myers, Optimal Financing Decisions, (Englewood Cliffs, N.J.: Prentice-Hall, 1965).

28. Salazar, R.C., and S.K. Sen, "A Simulation of Capital Budgeting Under Uncertainty," Management Science, (Dec., 1968), pp. 161-179.

29. Singer, H.W., International Development: Growths and Change, (N.Y.: McGraw Hill, 1964).

30. Truitt, J.F., "Expropriation of Private Foreign Investment: A Framework to Consider the Post World War II Experience of British and American Investors," (Dissertation, Indiana University).

31. Van Horne, James C., Financial Management and Policy, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1971).

32. Van Horn, R.L., "Validation of Simulation Results," Management Science, (Jan., 1971), pp. 247-58.

33. Weston, J.F., and E.F. Brigham, Managerial Finance, 4th edication (N.Y.: Holt, Rinehart, Winston, 1972).

34. Zenoff, D. and W. Zwick, International Finance, (Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1969).

Tutorial 2: GASP PROGRAMMING PROCEDURES
Chairman: A. Alan B. Pritsker, Purdue University, West Lafayette, Indiana

GASP II consists of a set of FORTRAN subprograms organized to assist the systems analyst in performing simulation studies. GASP II formalizes an approach to simulation by specifying common elements of simulation studies and providing subprograms for performing those simulation tasks that are independent of a particular problem. In this tutorial, the structure and subprograms of GASP II will be presented. Examples of the use of GASP II will be presented, which will include the programming required, the input data formats and the resulting summary reports. An introduction to GASP IV, a combined discrete/continuous simulation language will be presented.

Tutorial 3: SIMULATION OF ECONOMETRIC MODELS
Chairman: B. F. Roberts, University of California

The purposes of this tutorial are to present some fundamental properties of econometric model structures relevant to effective understanding of simulation procedures, and to describe the use of simulation software for analysis and forecasting.

The fundamental concepts of model specification are developed formally to provide a logical foundation for model construction and simulation procedures. The concepts of integrated and causal structures, reduced and final forms are examined and related to issues of: empirical identificatio and statistical estimation; interpretation; and computation. Procedures for decomposition of causal systems into ordered subsets of equations are outlined. The formal analysis is supplemented with specific examples.

The dynamic simulation capabilities of several available simulation software systems are described in general terms and specific procedures for loading and simulating econometric models with the California Economic Forecasting Project/Interactive Model Simulator (CEFP/IMS) system are given. Procedures for generation of dynamic multipliers, incorporation of judgment in forecasting and forecast error analyses are discussed. The session is concluded with an on-line demonstration of the CEFP/IMS.

Session 10: Transportation Models
Chairman: Richard de Neufville, Massachusetts Institute of Technology

The four papers in this session cover a wide range of transportation uses and problem areas. Thus, they are representative of simulation activity in the field of transportation, the potential of simulation in both large scale and small scale problems, and the application of different simulation languages.

Papers
"Simulation Analysis of Marine Terminal Investments"
David W. Graff, Esso Mathematics & Systems, Inc.

"Simulation in the Design of Unit Carrier Materials Handling Systems"
W. Wayne Siesennop, University of Wisconsin,
Fritz Callies, Rex Chainbelt, Inc., Neil S.Campbell, A. O. Smith Company

"A Generalized Model for Simulating Commodity Movements by Ship"
John C. Rea, Pennsylvania State University,
David C. Nowadling, University of Tennessee and
Philip W. Buckholts, R. Shriver Associates

"Simulation of Garland, Texas Vehicular Traffic Using
Current and Computed Optical Traffic Settings"
Frank P. Testa and Mark Handelman, IBM Corporation

SIMULATION ANALYSIS OF MARINE TERMINAL INVESTMENTS

David W. Graff
Esso Mathematics & Systems Inc.
Florham Park, New Jersey

Abstract

A common problem in the oil industry is the optimization of terminal facilities to minimize delays in servicing incoming tankers. In Exxon Corporation, simulation has been successfully applied to marine terminal studies since the early nineteen sixties. The development of a general model in 1967 contributed to wider use of marine terminal simulation throughout the company. This paper discusses the marine terminal investment problem, the basic technical features of this model, and a typical application of the model.

I. THE MARINE TERMINAL INVESTMENT PROBLEM

The oil tanker is a fundamental means of transportation in the petroleum industry. It follows that marine terminals, where tankers can be loaded or unloaded, are fundamental to a transportation system based on tankers. A refinery marine terminal is illustrated in Figure 1. The ability of an oil company to utilize its tanker fleet is dependent upon that firm's configuration of marine terminals. Anytime that a tanker is delayed in port, its capacity is lost to the transportation system. Accordingly, additional tankers must be brought or chartered to compensate for such delays. In planning facilities for a marine transportation system, it is necessary to estimate and plan for the time lost to port delays -- as well as to eliminate as much of these delays as is economical.

This problem may fall to the manager responsible for the overall transportation system

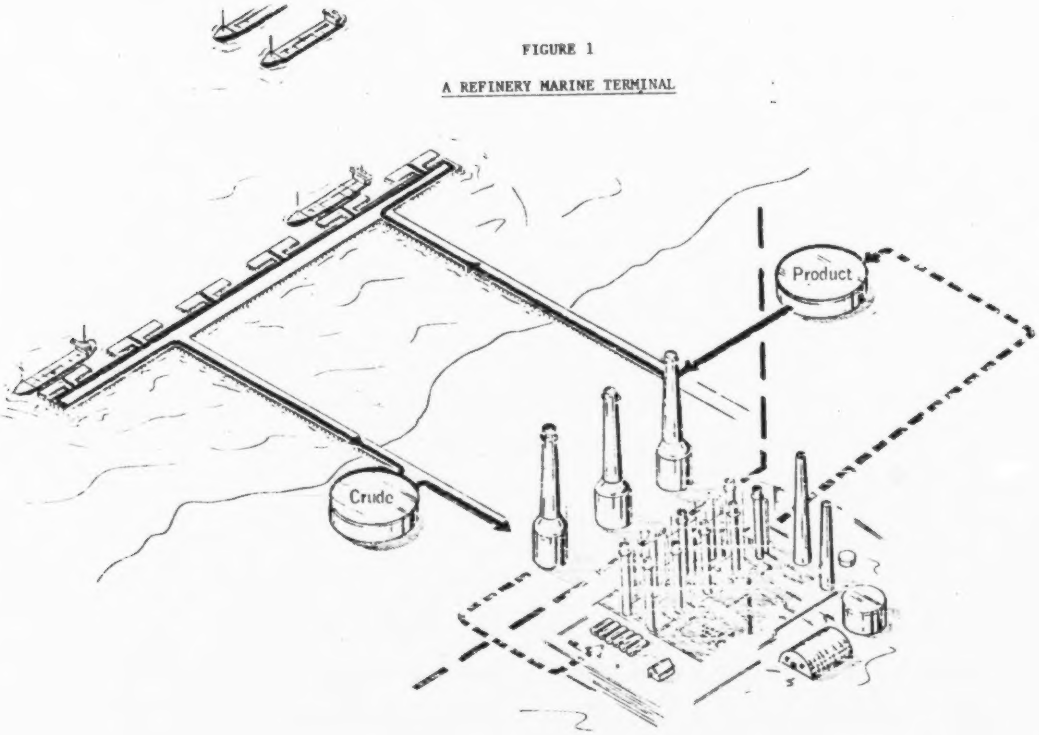


FIGURE 1

A REFINERY MARINE TERMINAL

or to the manager of a particular terminal in the system. Several kinds of investments can affect the service of a marine terminal. An increase in the size or number of tanks can reduce delays due to insufficient available capacity to unload the ship. Additional berths can permit the servicing of more ships at one time. On the other hand, one might achieve the same results by increasing the flexibility of one or more existing berths. In addition, it is possible that improvements can be affected by changing certain operating procedures, such as the rules for assigning ships to berths. More often than not, delays can be reduced most effectively by implementing some combination of the above alternatives.

The evaluation of these alternatives is quite complex. One must analyze the effect of different terminal facilities under various modes of operation. Furthermore, ships arrive with varying degrees of randomness, and service time is highly dependent upon the status of the system at the time of arrival. If ships arrived in regular intervals, then the system could safely be designed to process the average number of ships in port. Generally, however, the ship arrivals are "bunched", and the number of ships in port at any point in time may be several times the average. The effect of bunched ships competing for the same facilities can often be the most important factor in ship processing time, and this factor cannot be evaluated using average value analysis.

Proper analysis must take into account the complexities of current operations, but it must also anticipate the changes which will influence their operation in the future. As in any industry, factors such as processing volumes are bound to change. In addition, however, the oil industry is in the midst of altering the entire complexion of its marine transportation. For instance, many of the new tankers are extremely large. These ships bring more cargo into port at one time. They may take up more than one berth at a terminal. Voyages of these tankers are often restricted to specific routes, with smaller tankers transhipping cargo from large terminals to smaller terminals. New modes of operation are evolving in order to deal with the interaction of these new factors. As a result of all this change, it has become increasingly difficult to draw conclusions from intuition and past experience.

II. SIMULATION AS A TOOL FOR ANALYSIS

It should be evident that simulation is particularly appropriate for analyzing the operations of a marine terminal. Simulation has the flexibility to study this complex situation without imposing unreasonable simplifications on the problem statement. By the use of case studies, it is possible to analyze the impact of changes in facilities or operating procedures. Simulation can represent irregular and uncertain phenomena in the system. Furthermore, additional case studies may be used to measure the sensitivity of the system to changes in the projected operating environment, including such factors as demand, weather, and unscheduled maintenance.

Any marine terminal simulation model must realistically approximate two complex phenomena -- the arrival of ships and the servicing of these ships once they are in port. In order to simulate the arrival of ships, one must account for the influence of variability in the ship arrival patterns. The servicing of ships is best simulated by a detailed representation of the actual decision process which governs the operation of the terminal.

This kind of model requires a detailed set of input, describing vessel characteristics, product and crude demands, berth capacities and flexibilities, environmental conditions (such as weather and tides), and operating rules (governing berth and ship assignment). The output reports from a case can include both summary and detailed information on delays and inventory levels. In addition, the model can calculate the cost associated with the delays.

Delay costs are derived from the cost of chartering lost tanker capacity at projected market rates. As in any investment study, the terminal manager must evaluate any proposed investments against the related savings in projected costs.

III. DESCRIPTION OF A GENERAL MARINE TERMINAL SIMULATION MODEL

In the early nineteen sixties several models were successfully developed and applied to marine terminals in such places as Italy and Libya. Although each application more than paid for itself, steps were taken to reduce the time and money required to complete a particular simulation study.

Accordingly in 1967, a general model was developed with the specific design feature that it be easily tailored to most refinery marine terminals in the Exxon Corporation circuit. Existing technology was consolidated in this one model, and improvements in technology since then have also been incorporated in the model.

The model will be described from two points of view: the problem characteristics modeled, and additional technical features of the program.

A. Problem Features

1. Tanker arrivals at the terminal reflect a mix of planning and variability. The total number of arrivals each year is kept consistent with the total amount of crude or finished product processed during the year. The planned arrival time for each tanker is ideal from an inventory control standpoint, which reflects the actual tanker scheduling procedures. That arrival time, however, is subject to variation by inputting either a histogram or a standard deviation to the normal distribution.
2. If storms can close all or part of a terminal, the model will generate a pattern of storms of length and severity corresponding to the input provided by the user.
3. As each vessel arrives at the terminal, it is placed in the queue. The queue is ordered on a first - come, first - served basis, unless the user has elected a special priority basis. The special priority option groups the vessels according to priority class, then according to size within each priority class.
4. The complex berthing rules are summarized as follows:
    a. The berths are examined in preferential order, which is generally from least to most flexible. If a berth is free, and if weather permits, each ship in the queue is examined in turn until a match is obtained.

    b. The berth must be able to handle the size and cargo of the ship under consideration, and the lines required to load or unload the ship must be available at the berth.

    c. If the ship is to unload cargo, the ship will not be berthed until there is enough space in the tanks to accept the ship's entire contents.

   d. A berth might be "reserved" for an incoming large tanker. This would prevent a smaller ship from berthing if that would delay the large tanker.

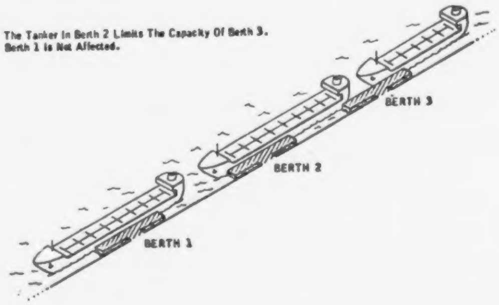
5. A ship which has qualified for berthing will go through the time delay required to maneuver into the berth and await line assignment, to complete the loading or unloading of its cargo, to release its lines, and to maneuver out of the berth. These components of port time are constant, provided facilities are available.

6. If the assignment of a particular ship to a particular berth should interfere with the capacity of a nearby berth, the capacity of the nearby berth will be adjusted during the time that the first berth is so occupied. This phenomenon is illustrated in Figure 2, where the large ship in Berth 2 limits the size of the ship which can go to Berth 3.

7. Crude inventories can be drawn down on a continuous basis. In the case of crude runouts, the crude drawdown rate is later increased until the loss is made up. Some tanks may be restricted to holding one crude only, whereas others may be available for use by several crudes.

8. Produce inventories are monitored. It is assumed that products are produced at a constant rate, while export or import operations are carried out by the vessels. The model makes no attempt to relate crude operations to product operations, or for that matter to relate the operations of different products to those of each other. To do so would require modeling the detailed refinery operations, which is beyond the scope of this model.

Figure 2



The Tanker In Berth 2 Limits The Capacity Of Berth 3.
Berth 1 Is Not Affected.

B. Additional Technical Features

This section describes certain technical features not specifically related to the problem structure of the model.

1. The model is coded in highly modular fashion. Input operations are in one location, output in another. There is a separate routine for each operation in the simulator (such as queuing, line connection, and berth departures). Consequently, it is very easy to isolate those portions of the program which require alteration in a particular study.

2. The model is programmed in FORTRAN. FORTRAN was selected because of its computing speed, its wide use throughout the Exxon Corporation affiliates, and because it lends itself very well to a highly modular structure.

3. The model contains an option to generate random numbers for ship arrivals and storm statistics using a "variance reduction" procedure known as "random sequence sampling". The procedure selects without replacement from a predesignated set of numbers, but in random order. The mean of resulting statistics (e.g., delays, turnaround time) is unaffected, but the variance is substantially reduced. In most cases, the mean of a statistic is the measure desired, and equilibrium conditions can be reached much sooner using this option, which thereby reduces computer time per case.

4. The model accepts input data in "free form". This means that data is identified by the use of keywords, rather than by card number or card column. As an example,

   PRTANKERS BOUNTY SIZE 80 NUMPRODUCTS 2

   would be interpreted to indicate that the product tanker BOUNTY has a draft size of 80 and carries 2 products. This input system was included in 1971, and since then it has proven far more viable than the former system based on card columns. The new method is also more amenable to the use of remote teletype or cathode ray tube terminals.

5. The model tailors the dimensions of nearly all vectors and arrays to meet the specifications of each case. This recent feature has eliminated the substantial re-dimensioning (and accompanied debugging) that used to be a part of every study. It has also eliminated the wasteful tendency to over-estimate array size in order to avoid later redimensioning.

In the future, additional features will be added according to the two processes which have brought about modifications and extensions to date: technological advances and refinements developed for particular studies.

IV. A CASE STUDY EXAMPLE

This section makes use of an example to outline the steps essential to virtually any application of the general marine terminal simulation model. This particular example is based on an actual study, done for one of the Exxon Corporation refineries in the summer of 1971.

A. Study Objectives and Manning Requirements

The specific objective of the study was to evaluate alternative proposals for pier expansion and also for additional crude oil storage tanks. A broader objective was to provide the refinery staff with a model that could be used for similar studies at any future date. Accordingly, the study was manned jointly by refinery personnel familiar with computer programming and by a member of the central OR group. The refinery personnel provided the expertise in the local terminal operations, and the OR man provided the expertise in the general model. By the end of the study, the refinery staff was completely capable of using the adapted model without further outside assistance.

B. Modeling Considerations

One of the first activities in setting up a study schedule was the description of the physical problem in modeling terms. This description could then be compared to the features of the existing model in order to identify the modifications required to represent the refinery's terminal operations.

The basic problem structure was well-suited to the application of the model. Vessels arrived at the terminal, based on an ideal inventory control strategy, but subject to random variations. Upon arrival, a vessel would be berthed immediately, provided sufficient empty storage existed to receive the ship's entire cargo, and provided there were an empty berth equipped to receive the ship. Otherwise, the ship would be placed in a queue until those conditions were met. In addition, subsequent arrivals of higher-priority vessels could further delay the servicing of a ship. Records were kept on each ship's total turnaround time in port, together with a breakdown on the various sources of ship delay.

Characteristic of other applications of this model, some aspects of this specific terminal's operation had not been anticipated in the model's design. Accordingly, portions of the logic had to be changed to complete the representation of the terminal. It was here that the modularity of the model proved especially useful. Because of the ease of changing one portion of the model without affecting others, substantial logic changes were incorporated by altering about 300 out of 7500 source statements in 12 of 51 subroutines. The major changes are summarized here to illustrate the kinds of factors that hinder complete generality in this type of model.

1. Transshipment vessels discharged one grade of crude oil and picked up another grade for delivery at another terminal. This required coordinated scheduling between all vessels for the two crudes. Normally, the vessels for each crude would be scheduled independently.
2. A scheduled maintenance period closed the terminal for twelve hours each week. This scheduled closure could occur only after all berths were empty, and consequently it imposed a restriction against berthing a ship too soon before maintenance began.

3. The queuing rules were more complex than those originally programmed, in that they allowed for increasing a vessel's priority if it had been delayed beyond a specified period.
4. The interaction of berth capacities (as illustrated in Figure 2) was also more complex than in previous applications of the model. Several berths were so close together that the berthing of a large tanker in one of them restricted the capacities of the rest of them.

C. Validation of the Revised Model

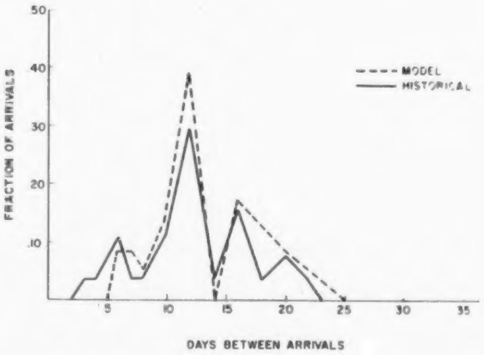
At the same time that the modeling changes were being specified and coded, data was being prepared for a validation case, taken from the refinery's 1969 records. Validation of the model consisted of checking model performance against actual results, and also of establishing how long the model should be run to represent system performance. Both facets of validation were prerequisites to using the model for case studies of future performance.

The most time-consuming part of validation consisted of comparing model statistics with historical results. As a first step, the model demonstrated that it could generate vessel arrival patterns representative of those during the 1969 test period. Figure 3 shows one comparison between the historical distribution of interarrival times and the corresponding distribution generated by the model. Secondly, once the arrival patterns had been validated, the model was also able to produce realistic operational statistics for berth occupancy, delays, and maintenance.

In addition to validating the model against past performance, it was necessary to determine the length of simulated time required to achieve equilibrium. This was established by comparing the results of a particular case run several times using ship arrival patterns generated from different sequences of random numbers. In this instance, the results become stable after four years of simulated time. This required approximately seven minutes of 360/65 CPU time per case.

FIGURE 3

HISTORICAL AND SIMULATED INTERARRIVAL DISTRIBUTIONS FOR CRUDE A TANKERS

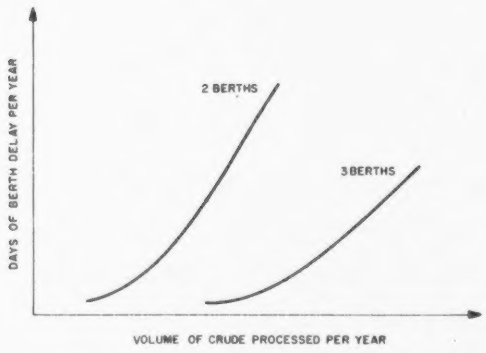


D. Case Studies

Case studies and analyses were completed following the validation of the model. Figure 4 illustrates one set of relationships established by running case studies with the model. This particular graph relates the volume of crude processed per year (which dictates the volume of ship traffic) to the expected time spent waiting for berths. Curves have been plotted for configurations of two and three berths. The decision to build the third berth would, of course, be justified when the volume reached a level such that the (time-adjusted) difference in delay costs exceeded the investment outlay. Case studies would also be used to derive similar relationships for crude storage tanks or other terminal facilities.

FIGURE 4

EFFECT OF REFINERY VOLUME ON BERTH DELAYS



E. Study Duration

Model re-design, revision, validation, and the running of initial cases were carried out in nine weeks. At the end of that time, the development was complete, and the refinery staff were completely indoctrinated in the application of the model. No further outside support was necessary either to run additional cases or even to modify the model if the need arose.

V. CONCLUDING COMMENTS

For the past decade, simulation has proven to be an effective tool for the study of marine terminal facilities. The general model has provided numerous studies with a framework for problem identification, solution, and analysis. There is every expectation that marine terminal simulation will continue to be a widely accepted technique throughout the affiliates of Exxon Corporation.

SIMULATION IN THE DESIGN OF UNIT CARRIER MATERIALS HANDLING SYSTEMS

W. Wayne Siesennop
Systems-Design Department
University of Wisconsin-Milwaukee
(formerly of Rex Chainbelt, Inc.)

Fritz A. Callies
Technical Center
Rex Chainbelt, Inc.

Neal S. Campbell
Data Systems Division
A. O. Smith Corporation

ABSTRACT

Unit carriers are used in satisfying many current materials handling needs. A system using this kind of car-on-track hardware can be extensive and complex. This paper discusses the values and benefits of simulation as applied to the design of such a track system. The design of the automatic baggage handling system for the Seattle-Tacoma International Airport is used as an example of such a unit carrier system. Design requirements include input rates of baggage loading stations and car destination patterns over a relatively fixed system geometry. A simulation model is used to evaluate overall performance including empty car availability and minimized network flow. Design of the trackage controls as well as tuning of the entire baggage handling system are aided by use of the model. The results show that simulation is a highly-cost-effective tool for this problem.

INTRODUCTION: A unit carrier is defined as a vehicle used to transport one or more items to a certain destination. In warehouses designed for automated order picking, unit carriers are loaded with several items and deliver these to a specific control point. In institutions and post office factories individual carriers or trays are used in a similar manner. Another common example is a passenger elevator. In a unit carrier system these discrete, individual vehicles carry an item or items along fixed routes, each relatively independent of the other vehicles in the system.

In the case of the unit carrier, all handling equipment interfaces with the standardized carrier rather than with the item carried. This is in contrast to conventional materials handling hardware where each item moves along a conveyor system with its position fixed with respect to other objects. The items interface directly with the conveyor belt chain or rollers and with any control or processing equipment of the conveyor system.

Unit carriers are particularly useful when the items are fragile or of nonuniform size or, in general, when they are difficult to handle by conventional conveyor methods. Because positive control as well as ability to remember are features of the unit carrier design, items can be processed at greater rates and with more reliability and safety than items on a conveyor belt.

Accompanying the advantages of unit carriers however, are the problems associated with the dramatic increase in the level of complexity and sophistication required. Unlike a continuous processing system, the typical unit carrier system contains a large number of relatively independent containers traveling to different destinations through a network containing branch points and merge points. The unit carrier system requires a supply of available carriers at the origin, and a place to store the carriers not in use. In order to design such a system, the control logic must allocate carriers to different areas within the network. The result is a complex track network requiring considerably more sophisticated design techniques than those required for conventional conveyor systems.

This paper concerns the use of simulation in the design of a unit carrier baggage handling system for the Seattle-Tacoma International Airport (1). This baggage system presents problems which are common to many unit carrier materials handling system applications.

SEA-TAC BAGGAGE HANDLING SYSTEM - Airport congestion, due to increasing air travel and larger aircraft, is becoming increasingly familiar. Aircraft congestion slows arrivals and departures. Automobile congestion and limited parking facilities are increasingly troublesome. With this added traffic and congestion have come the problems of handling larger quantities of baggage which must be moved quickly and carefully in ever-increasing volumes, while minimizing damage, pilferage, or mis-direction.

Currently, most airport terminals handle baggage by means of well-designed conveyor systems and ramp vehicles. But for a growing number of airports, particularly those of medium and large size, these are no longer adequate. Unit carrier systems provide a highly flexible, high-capacity and high-quality solution to these baggage handling problems. Several companies now produce such carrier systems for this baggage handling market (2) (3).

The new Ground Transport Express (GTX) system at the expanding Seattle-Tacoma International Airport involves over 1,000 four-wheeled carriers (Fig. 1) operating on an extensive network of 20,000 feet of track and serving some 3500 peak hour passengers (Fig. 2). The track network, which has 80 switch and merge points, connects all of the airlines, including two satellite terminals and four parking lot check-in stations, into one unified system. It facilitates check-in and interairline transfers.

A typical GTX carrier trip is represented schematically in Figure 3. A carrier moves to a loading position from an adjoining storage line. It is loaded with luggage and coded for the proper destination. The carrier then moves through the complex track network to the proper dump station. After discharging the baggage, the empty carrier seeks the nearest available storage line.

In the Sea-Tac system, baggage enters the system at passenger check-in counters in the main terminal and in the parking lot. It proceeds by carrier to different luggage sorting areas where the baggage handling system automatically sorts it by airline or, in some cases, by flight number. The baggage is then transported by ramp vehicle to the airplanes. Inter-and intra-airline luggages is also handled by the GTX system, although at Sea-Tac, deplaning luggage is sent by conventional methods to the baggage claim area.

The unit carrier in the GTX system is a 185 lb. car with a V-shaped pocket formed in the body to provide a secure holding place for almost any size and shape of passenger luggage (Fig. 1). Motive power is supplied to the car by means of a friction drive mechanism built into the track. A probe on the carrier allows it to switch from one track to another upon command. The detailed physical characteristics of the hardware of the carrier system for the Sea-Tac Airport are described in (4).

Specialized equipment allows cars on secondary lines to merge into traffic on a primary line. Inclines, declines, and vertical lifts allow for necessary changes in elevation. The carrier is capable of dynamically discharging its baggage onto a stationary slide or moving belt collection system.

Carrier traffic is controlled by means of magnetic sensors located at decision points along the guideway (5). These sensors interrogate a memory, fixed to each car, which carries a binary code indicating the empty or full condition of the car and its destination. This code is entered at the loading point. After discharge, the memory is changed to indicate an empty carrier.

The system, then, must manage a large number of independently programmed carriers operating simultaneously, traveling from a large number of origins to a large number of destinations on a network of track which is constrained both by cost and by the existing physical track configuration environment. The system must operate efficiently under all circumstances and, in fact, its primary justification is its lower per-unit cost at both low and high activity levels.

SYSTEM DESIGN AND ANALYSIS - An acceptable system must be designed to at least meet the customer's specifications. Regarding baggage movement at the Sea-Tac International Airport, two major flow characteristics were specified:

1. Travel Time - this is a measure of the maximum amount of time required for a bag to travel to its destination in the system. This includes the time needed for an empty carrier to travel to the check-in station, to be loaded, and to travel through the system.
2. Peak Network Flow Rate - this rate specifies the number of bags per minute that each originating station may send to each destination under peak conditions.

These two criteria, in turn, imply other constraints on the operation of the system. For example, the ability to handle a certain peak number of bags per minute from a certain facility implies that there must always be an adequate supply of available carriers at that facility. The maximum travel times from point to point imply that the traffic densities at peak conditions must not be so large as to prevent the merging of traffic at critical points in the network. The restriction on traffic densities at certain points implies that unnecessary movement of empty carriers should be minimized to speed the flow of the full carriers through the network.

Any unit carrier system can be designed with enough redundancy to insure its satisfactory performance under any circumstances. The cost of redundancy, however, is high. It may mean extra elevators, fork trucks, or in the GTX system, carriers and track. The problem, then, is to produce a system which will meet a specified set of performance criteria under anticipated operating conditions with a minimum of redundancy, excess capacity, and unnecessary activity.

The design problem involves finding a way to relate the system's measures of effectiveness to the system's design parameters to allow meaningful cost-benefit trade-offs to be examined. For example, a lack of empty carriers available at a specific loading area may be due to the system logic, the total number of carriers in the system, the physical track configuration, or other aspects of the system. The analysis and design problem is to relate the measure of performance, such as the availability of empty carriers in a particular area, to a system design parameter such as the total number of carriers in the system, in order to allow the designer to appreciate the effects of design changes on performance.

The design of a dynamic system typically has two major phases. The first phase is a static or steady-state phase; it essentially asked what the maximum loading conditions are like, on average, and uses this average loading to produce a preliminary design. This phase can use the conventional methods of analysis much like those used in balancing assembly lines. Assuming that the system must be able to operate at steady-state at some maximum rate, the capacities of each part of the system can be determined from these input rates. This is really a mass-flow analysis: cars in equals cars out.

In the past, this static analysis has been sufficient for many systems. Tuning of the system was done by adjusting either the real system after it was built or an actual physical model. This tuning corresponds to the second major phase of systems analysis and design. Using the preliminary design, the second phase investigates system response to dynamic loading conditions. This dynamic analysis deals with the system response to such factors as random elements in the loading, system start-up, system shut-down, or a net cross-flow of carriers in the system. The random nature of arrivals to to the airport may for short periods transform a low average demand rate per hour to a demand well in excess of the design criteria. Rapid start-up may cause temporary shortages of empty cars. Certain random loading conditions may produce an unbalanced cross-flow of carriers in the system. This can result in shortages of empty carriers if the control logic does not respond adequately to compensate for this flow condition. Several of these conditions may occur at once, compounding the problem.

More than the conventional static design tools are required to intelligently design for this kind of system loading. The usual continuous modeling techniques used in many engineering control system design applications do not allow for probabilistic inputs. Queueing models allow for certain kinds of random elements in the system but become unmanageable for large systems. In addition, the queueing theory assumptions often become unrealistic and confining for such problems, and formulation of queueing model segments require data which are not always readily available.

Use of analytic models requires that the problem be divided into small components which can be dealt with and for which clear relationships among design variables can be defined. Analysis relying entirely on these small units may lead to a system in which each component is designed independently. This can result in good component design and poor system design. Optimizing subsystems does not necessarily optimize the total system.

To cope with such large system problems, computer simulation is regarded as the most cost-effective tool. Use of a computer simulation model allows the designer to gain experience with the system as the system design and model evolve. He then has a clearer understanding of system performance relationships while the design process is still going on. In this way the designer's understanding of the significance of different measures of effectiveness can guide the progress of the design by providing fast, efficient feedback as changes in the system are made.

A simulation analysis usually has credibility not only with the designers, but also with management and decision makers. The manager can be shown what the computer is doing at each stage of the process, and the manager can verify that the computer model does represent the process being modeled. The model can be coded in as much detail as is necessary to reflect the real system, without worrying about fitting the constraints of a particular theoretical model. The simulation approach allows examination of the effects of many different and unusual loadings. Control parameters and alternative configurations are readily changed. System design then becomes an iterative process with the model builder searching for the best solution in terms of his evolving understanding of the problem at hand.

SIMULATION PROJECT PLANNING AND MANAGEMENT : With an iterative design process for a large system, project planning and management are critical factors. A lack of careful planning and management can allow project costs to become greater than necessary without significantly improving the results. The tendency to include more detail than necessary can increase data collection, model development, running, and data analysis costs. A model structure which does not take advantage of system modularity can result in higher development, testing, validation, and running costs. Poor choice of a simulation language can result in higher model development costs or running costs. With these problems in mind, the GTX model was carefully planned and budgeted.

For the Sea-Tac System, the possible project conclusions included the following:

1. Confirming that the original system design was satisfactory
2. Revealing that the original design, or portions of it, was unworkable
3. Discovering control logic changes and additions which would improve system performance
4. Finding ways of cutting costs without impairing system performance

Each of these outcomes was valuable, and each was present in some degree. The combined value of these results was the value that was relevant to the development of the project development and design.

Based on a preliminary examination of the project as a whole, a specified task, a specific completion date, and limited resources were assigned to the effort. The project design was thus controlled by both available resource inputs and the required outputs.

The next step in the planning process was to allocate the available resources to produce the desired results. The demands to be placed on the model and the inputs to the model were defined. How the model would be used and what kinds of experiments would be run were important design considerations. The resulting design stressed simplicity of output and modeling, modular design, and testing of modules prior to assembly of the total model. Associated with the modular design was a project plan with milestones and review points. These were important factors in the economic as well as design success of the project.

ACTUAL MODEL - Having analyzed the problem and established the project budget, completion time and environment in which the model was to operate, the final level to consider was the model itself. One of the most important aspects of any model planning is the choice of a simulation language. The language chosen for modeling the baggage handling system was General Purpose Simulation System (GPSS). This choice was made in order to reduce model development time at the expense of somewhat longer running times and larger core requirements. It was anticipated that model development would be the most expensive and time-consuming portion of the project. With discrete carriers flowing through a network of tracks, this system was a good application for GPSS, which is oriented toward flow type systems.

For this baggage handling system, the system geometry was fixed (Fig. 2) before the simulation was initiated. The total baggage handling system network, consisting of about four miles of guideway, was divided into five subsections. Each section was programmed, debugged, and tested independently and then the segments, which are too highly interactive to give meaningful results independently, were joined into one total system. This modular construction minimized the cost, time, and computer charges required for model programming, testing, and debugging.

The design choices were assumed fixed for such design parameters as number of carriers in the system, line speeds, and storage bank capabilities. The latter was largely influenced by architectural and constructural considerations. Therefore, the model was not organized for easy alteration of these parameters.

Speeds and distances were converted to delay-times for each piece of equipment (lifts, inclines, turntables, lines, etc.) involved in the system. These were grouped into about 140 line segments, each with the appropriate transit time (Fig. 4). Attempts to model randomness in these transit times were a refinement deemed unnecessary and one that would complicate debugging and checking of the simulation output.

With this planning completed, the actual coding began. The Sea-Tac system model, which contained about 1500 GPSS statements, was developed in an elapsed time of two months, and represented about a three man-month effort. This included time required to plan the project, collect data, build and test the modules of the system, and asemble the complete model. Then one month was spent in an iterative process revising both system and model logic to achieve a reasonably satisfactory level of operation. At this point the simulation could be considered a finished tool representing an operable system. An additional five months were then spent using this tool to develop and refine the logic of empty car control, to evaluate the effects of specific control hardware, and to work out solutions to problem areas. Working with the simulation was a great aid to imaginative innovations; however, the project time and budget constraints regulated the degree to which these could be pursued.

The simulation model, once constructed, was used to understand the system by asking "what if" questions. Using different rates of baggage arrivals at different parts of the network, the response of the system was carefully monitored. The number of cars in each section was monitored, as was the rate of flow of traffic at critical points, the availability of empty carriers, and the cross-traffic from one section to the other. The system was examined during system startup in the morning, peak operating levels during the day, and as the system activity decreased later in the evening. Arrival rates were changed at different sections of the system to test the ability of the system to respond to uneven loading in various parts of the network. Each of these experiments served to provide a deeper understanding of the system's performance, and point out critical areas in the system.

Some of the relatively fixed design features were changed as the simulation pointed out problem areas which needed to be corrected. In most cases where changes were originally anticipated, the input was programmed for easy changes. Control logic at each switch was represented by a predefined true-false Boolean variable, a set of logical conditions which, when satisfied, cause the carrier to switch. These Boolean variables evaluated such things as whether the carrier was loaded and, if so, its destination. For an empty carrier, the levels of relative need in various storages would influence its path.

The generators which created GPSS transactions to represent the arrival of baggage were controlled by values which could be initialized or changed at any predetermined time in the course of a simulation run to reflect various system loadings. Destination of the carrier from various input locations was controlled by random number generators which would produce distributions to match predetermined but easily changed functions. Thus the system was represented by a mathematical model including both the internal system logic and the inputs arriving from outside the system.

The results of the simulation are no more accurate than the inputs to the simulation. Therefore data must be gathered or generated with great care. In the case of the Sea-Tac simulation, a complete engineering study was conducted prior to the simulation (7). Baggage input spectrums were derived from that report and from the customer specifications. Figure 5 is an example of the magnitude and distribution of total airport baggage arrivals used for input data in the computer simulated system.

With clear measures of cost effectiveness available, the required level of detail in the system was initially established. More detail was added later in the project, but only after it became clear that the added detail would have a significant effect on the system performance at key points in the network. For example, it was initially decided that all merges would be simulated as working without physical constraints. When areas were detected where rates were exceeding equipment capabilities, merge suppression was modeled. By adding this detail only at key points, however, significant costs were avoided without jeopardizing the project results.

MACRO SUBROUTINE EXAMPLES - Exploiting program modularity through the use of GPSS MACROS, or subroutines, further reduced coding and debugging time and expense. Macros used in the system simulation were designed to standardize and document the modeling of similar physical situations, and provide the basic framework around which the rest of the system model was built.

The proposed system of releasing empty cars from storage banks was modeled in a macro; eventually several different macros were used to represent the different methods of ordering cars. Macros were also programmed for each kind of station, carrier storage queue, and other facilities which occurred at several points in the network, as well as for various initiation routines. These macros were then called at the appropriate point in the program. A discussion of one of these macros follows, to illustrate this capability.

The Load Macro (Fig. 6) describes a portion of the system where bags are loaded into carriers. This simplified diagram, in which each symbol represents one GPSS statment, illustrates the one-to-one relationship between GPSS statements and system logic. The macro models the process of loading empty cars and initiating calls for additional cars to be released for loading if there are both empty cars and more bags waiting.

In the Load macro (Fig. 6) empty cars enter the loader and occupy the loading facility A. The car, represented by an GPSS transaction, then moves into two successive ASSIGN blocks which store the number of the origin B and the number of the destination C in parameters associated with the car. The transaction, the car, then moves into an UNLINK block which removes one bag from the bag queue D and loads it into the car.

Having performed these bookkeeping functions in zero simulated time, the transaction then enters the ADVANCE block where it spends the loading time E (typically 3.7 seconds for a station rate of 17 cars/min.). When the loading time has elapsed, the transaction enters a TEST block and tests whether or not there are empty cars waiting to be loaded and bags waiting in the queue. If this is true, the transaction moves into the second UNLINK block and releases one empty car from the storage bank H. If there are no bags waiting to be loaded, or no empty cars available to be sent out, the transaction goes to the RELEASE block G. It leaves the loading facility A, and the transaction representing the full car leaves the loading area.

OUTPUTS IN RELATION TO GTX SYSTEM DESIGN - The philosophy of adding details only where required was followed in specifying simulation output. Initially the only output was the standard GPSS statistics. Once the model had been debugged and verified, it furnished a broad overview of the overall performance of the system. In addition, it provided the ability to examine in any desired depth of detail the areas of special concern. As key problem areas in the system became apparent, additonal tables were added.

Problem areas common to unit carrier installations have been addressed previously (6) and determine overall system effectiveness. Certain of these problems, including excessive line flow rates, insufficient empty car supply, and overall system imbalance, were examined in the Sea-Tac system design. System imbalance can result from many occurrences, including high activity within one particular area or a disproportionate number of loaded carriers sent from one station in the system to another. Both have the tendency to deplete certain areas of empty cars, resulting in lack of containers in thoses areas and causing high line desnities in other sections. Lack of empty cars at a check-in area is one of the most serious system problems. If empty containers are absent, the airline passenger is delayed and the airplane may be detained. This not only results in a poor customer relationship, but may also mean additional expense for the airline. It was a prime design consideration.

One of the primary indications of the performance of the baggage handling system was the length of the baggage waiting queues. Baggage waiting queues formed at loading stations when the system response was such that an insufficient number of empty cars was available to handle the rate of baggage arrivals. Figure 7 shows the amount of excess baggage that was not removed from certain stations under the baggage input conditions given in Figure 5. However, as the system responded and sent the necessary empty cars to those stations, the baggage waiting queues decreased. Important variables shown are the amounts of baggage in the queues and the time it takes for the queues to be relieved.

For a better understanding of why the system handled or did not handle baggage in various areas, it was necessary to study the availability and movement of the empty carriers. Several types of output contributed to this analysis, the most important being the tabulated conditions of empty carrier storages (Fig. 8). Shown in the figure is the activity of each empty car storage area in the system, including a number designating each storage and the empty car capacity for that storage, current contents of empty cars, and the rate of cars entering the storage at this particular point in time.

In general, a low level in a storage bank at any given time is not always significant. Perhaps no carriers are needed in a certain storage during normal running conditions; this is the case for ST080. Maybe the carriers needed are already enroute. Likewise, a moderately high level could still be dangerously low. In the case of long tunnels, a number of carriers may have entered previously and not yet emerged. Therefore they are not available for immediate use. Or, perhaps an inadequate flow may be presently arriving which may cause a future shortage. The average level over some small period is more meaningful.

A second measure of empty car availability can be used. Other tables exist which show the number of carriers in line at the exit of the storage area. But again, a low level is not necessarily significant. If a carrier always arrives just when required, it will never stay in the storage, so the number of carriers stopped and waiting could be small even with a large number of carriers enroute. The problems inherent in these two methods of counting empty car storage contents resulted in a detailed study of the type of electrical sensors to be installed at various points in the physical system and their effect on system logic.

Other graphical portrayals help analyze the condition of critical storages over time. Figure 9 is one example showing the activity in the north tunnel bank, the main empty carrier reservoir in the north half of the GTX system. The return route from the North Satellite contains a storage bank line for 261 carriers and a high speed line which allows empty carriers to bypass this storage when they are needed immediately elsewhere. The figure shows that more cars than necessary were entering the storage and then immediately leaving, when in fact they should have been bypassing the storage altogether. In later simulation runs different control methods were employed to reduce this flow.

In order to analyze where the carriers were going to and coming from, it was necessary to study the block counts of activity through each logic block of the model. From these a plot of traffic density and carrier location distribution everywhere in the system could be prepared. This was of great value, particularly in analyzing traffic at merges.

Line flow rate is limited by the car velocities and by the processing rates of such equipment as inclines and elevators. If the line density becomes too high at the limiting speed, merges become bottlenecks. At merge points, cars on the secondary track are halted in a queue until an adequate gap occurs in the primary line to allow merging. The higher the flow rate on the primary line, the less opportunity for cars to merge, thereby resulting in a longer waiting line. Each holding position in the track requires some specialized hardware to stop, retain, and then advance the carriers In addition, when the number of positions in the queue is filled, the secondary line must shut down until the queue is relieved. The number of queue positions at a merge point is a matter of economics as well as geometrical restriction. Minimizing line flow rate minimizes merge problems. For example, the traffic density was such that it seemed advisable to add to the model a simulation of the merge at the output of the north tunnel storage bank. The efforts expended in simulating this area were also applicable to developing the actual merge control used in the physical Sea-Tac system.

Studies of traffic densities also pointed out other problems. Carriers were diverted by some preceding station or logic test before reaching their required destination, or they were trapped in a loop. The latter type of problem is illustrated with the loop shown in Figure 4. After the first run with one set of logic, it was noted that there was abnormally heavy traffic in the lines in this loop. A review of the logic for switches SWB02 and SWB05 revealed an inconsistency which was preventing cars from leaving the loop. After this was corrected, traffic was reduced to a reasonable level.

In addition, tables were specified to record the origin and destination of all carriers passing certain points. Fig. 10 gives the origin table ORB11 for line LNB11 and the destination table DEB12 for LNB12. Table ORB11 shows the origins of cars passing on LNB11 during a given time; the column Upper Limit indicates car origin station. Similarly, Table DEB12 gives the destinations of cars passing on LNB12, the column Upper Limit representing the destination stations, and Observed Frequency indicates the number of carriers passing through LNB12 destined for those stations. Destination zero indicates an empty carrier.

A more detailed study of destination and origin tables aided in locating GTX system control logic errors and in redesigning workable control logic. In an initial model run, these tables revealed that very few of the empty carriers required on LNB13 came from LNB12 (Fig. 4). Most empties entered storage bank 70, then left again on LNB23 as required. The result was a large number of carriers in dynamic storage, i.e., on their way into and out of bank 70, rather than bypassing via LNB12. A set of bypass logic was devised to prevent the carriers from going into this loop. Analysis of the effects of the new logic revealed that under certain circumstances it resulted in starving station 10. Consequently, a reserve level had to be established for storage bank 70, below which it could have priority. All these considerations were eventually integrated into a set of final logic which minimized undesirable system responses.

EMPTY CAR MANAGEMENT - Many of these problem areas are related to each other and are really symptoms of the overall empty car flow management problem. The trip of a full, coded carrier is deterministic in the sense that the path of travel is known, and the trip times can be estimated, within certain limits, depending on system traffic levels. However, empty car flow is much more complicated. Once the car has been unloaded, it is coded as empty and sent into the system to seek a home in an empty car storage line. The car may pass through a number of switches before reaching an empty car storage line.

When directing empty cars, the switches can be generalized as either area switches or storage line switches. An area switch allows the carrier to move into a particular area which contains a multitude of storage lines; a storage switch allows the carrier to enter a particular storage line (Fig. 11). When an empty car reaches an area switch, the switch directs it into either area A or area B, depending on the relative number of empty cars in storage in these respective areas and/or the priority of the one area over the other. After a carrier is in an area, it approaches a storage switch (A-1, Fig. 11). If storage A-1 has room and there are no priority needs downstream, the car enters. Otherwise it passes, traveling on to switch A-2.

Another aspect of empty car management enables certain empty car storage lines to call for empties when their level becomes critically low. If the level of empty cars in storage A-2 is low, a signal can be sent to storage A-1 to release a certain number of cars. Upon reaching switch A-2 these cars will satisfy the needs of storage A-2. Another form of signal can also be initiated when storage A-2 is low. The signal will block Switch A-1, preventing any empties from entering storage A-1. It may even block the area switch preventing any empty carriers from being diverted to area B. Therefore all empties will go into A-2 until its needs are met.

Figure 12 is a simple model of only a small part of the overall system. In reality, there are 39 storage lines scattered throughout the Sea-Tac installation with capacities ranging from two carriers to 276. Their placement and capacities are, like the track itself, subject to architectural constraints.

CONCLUSIONS: The Sea-Tac simulation was useful in several ways. It clearly showed that the static preliminary design alone was insufficient for a complex unit-carrier system, and that simulation was a very helpful tool in design of dynamic systems. This was demonstrated not only in an engineering sense, but also in an economic sense--doing the analysis quickly and at a relatively low cost. The modular design approach to development was a central part of this cost effectiveness. A set of macros was developed and then used extensively. In addition, building and testing the entire system in pieces and then assembling these saved many hours and dollars.

With this simulation tool available, it will be comparatively easy in the future to model similar systems with relatively small changes to the design approach.

From an engineering standpoint, the major result was a good final design. On a more specific level, it became clear that the symptoms by which the system's performance would be measured were all part of empty carrier management. This area could be systematically analyzed and workable solutions developed.

In designing the GTX system for empty car management, then, it became evident that:

1. A priority must be established among storage lines
2. The overall capacity of each line must be set within geometric constraints
3. Operating levels and critical levels must be determined for most storage lines
4. Depending on storage lines priorities and their desired performance parameters, the automatic call logic must be designed.

These points constitute the controlling elements of empty car management. They are highly interactive, and adjustments at any one point may have far-reaching ramifications throughout the network. Thus a computer simulation becomes a necessary part of the design process in order to develop the best logic system.

In some cases there is not one best logic. A perfect control system would have to anticipate demand changes ahead of time. Varying input of bags at different stations results in different requirements on the system. It is only possible to set relative priorites on the goals and to strike a compromise between such conflicting requirements as immediate car availability and minimum non-essential traffic. However, the simulation allows the designer to try various schemes and precisely monitor their effect throughout the system while other parameters remain constant, an accomplishment which may never be possible in the real life system.

REFERENCES

1. "Airport's $125-Million Expansion Squeezes Around Old Terminal", Engineering News Report, Feb. 18, 1971, pp. 26-31.
2. "Automated Baggage System Tested for Use by Airlines", Aviation Week & Space Technology, Oct. 20, 1969, pp. 150-161.
3. "Airport 'Roller Coaster' Gives Your Bags a Fast Ride", Popular Science, May 1971, pp. 34-36.
4. G. J. Eggert and W. W. Siesennop, "The Use of Destination-Coded Unit Carriers to Solve Airport Baggage Handling Problems", presented at the Joint IEEE/ASME Materials Handling Conference, Milwaukee, Wisconsin, Oct. 1971, IEEE Paper #71 CP 731-IGA.
5. G. J. Eggert and R. J. Patton, "Using an Array of Magnets for an Escort Memory", Automation, January 1972, pp. 50-51.
6. L. W. Hillman, "An Order Picking and Shipping Model", Proceedings of the Third Conference on Applications of Simulation, Los Angeles, California, December 1969.
7. "Engineering Performance Study Report, Baggage Handling System, Seattle-Tacoma International Airport", Seattle, Washington, 1969.

FIGURE 1

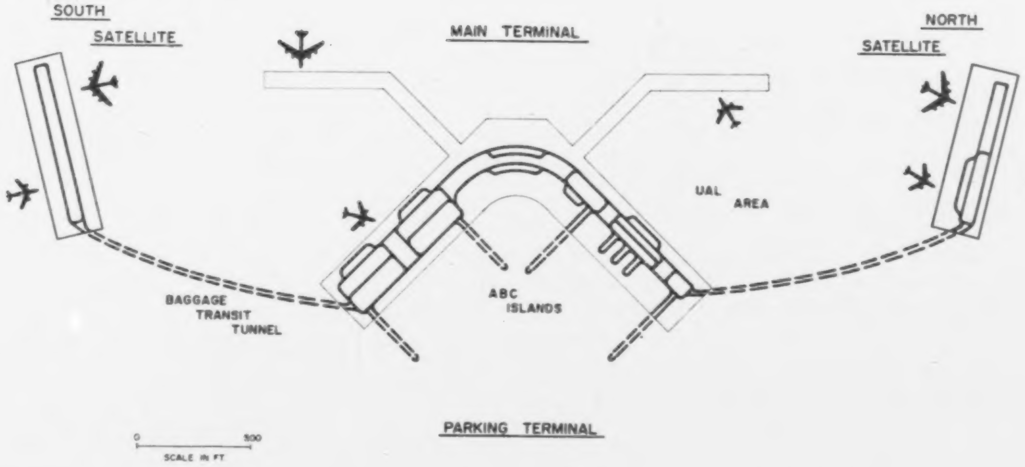

SOUTH SATELLITE
MAIN TERMINAL
NORTH SATELLITE
UAL AREA
ABC ISLANDS
BAGGAGE TRANSIT TUNNEL
PARKING TERMINAL
0
500
SCALE IN FT

FIGURE 2
Simplified Schematic of the GTX Baggage Handling System

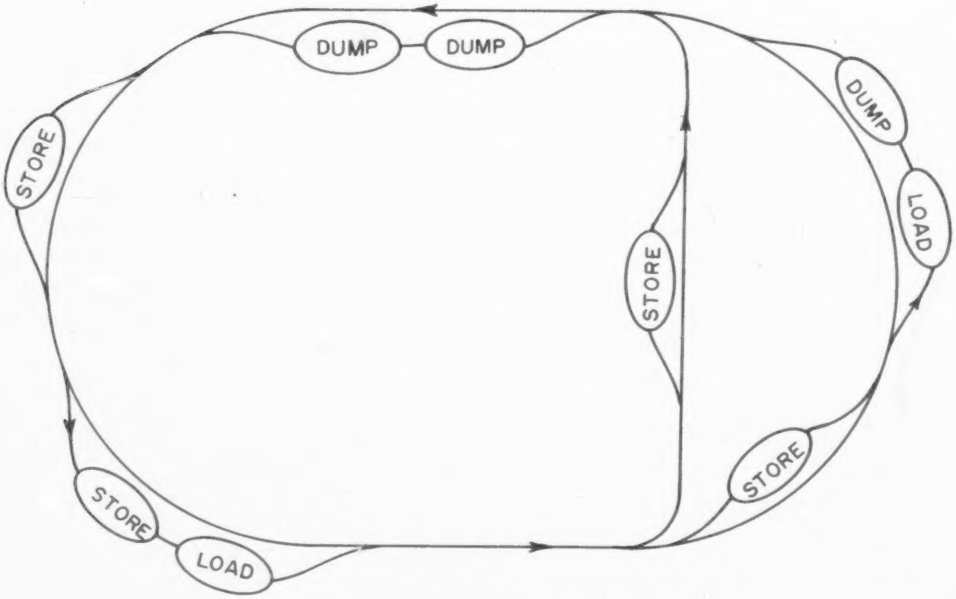

DUMP
DUMP
STORE
DUMP
LOAD
STORE
STORE
STORE
LOAD
STORE

FIGURE 3
Basic Carrier Network

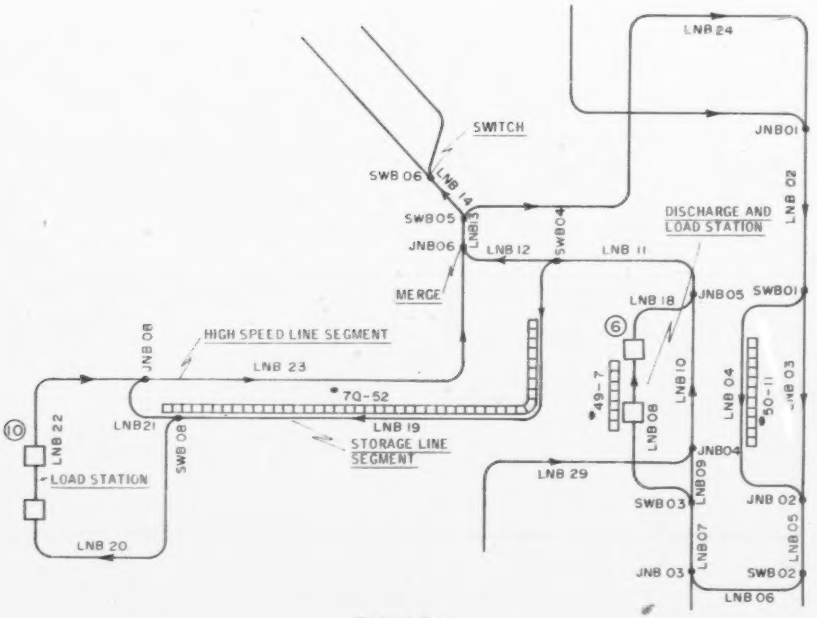

LNB 24
SWITCH
JNB01
SWB 06
LNB 14
LNB 02
SWB05
SWB04
DISCHARGE AND LOAD STATION
JNB06
LNB13
LNB 12
LNB 11
MERGE
LNB 18
JNB05
SWB01
⑥
JNB 08
HIGH SPEED LINE SEGMENT
LNB 23
49-7
LNB 10
LNB 04
50-11
LNB 03
7Q-52
LNB21
LNB 19
LNB 08
⑩
LNB 22
SWB 08
STORAGE LINE SEGMENT
JNB04
LOAD STATION
LNB 29
LNB09
SWB 03
JNB 02
LNB 20
LNB07
LNB05
JNB 03
SWB 02
LNB 06

FIGURE 4
Detailed Network of a Portion
of the Simulation Model

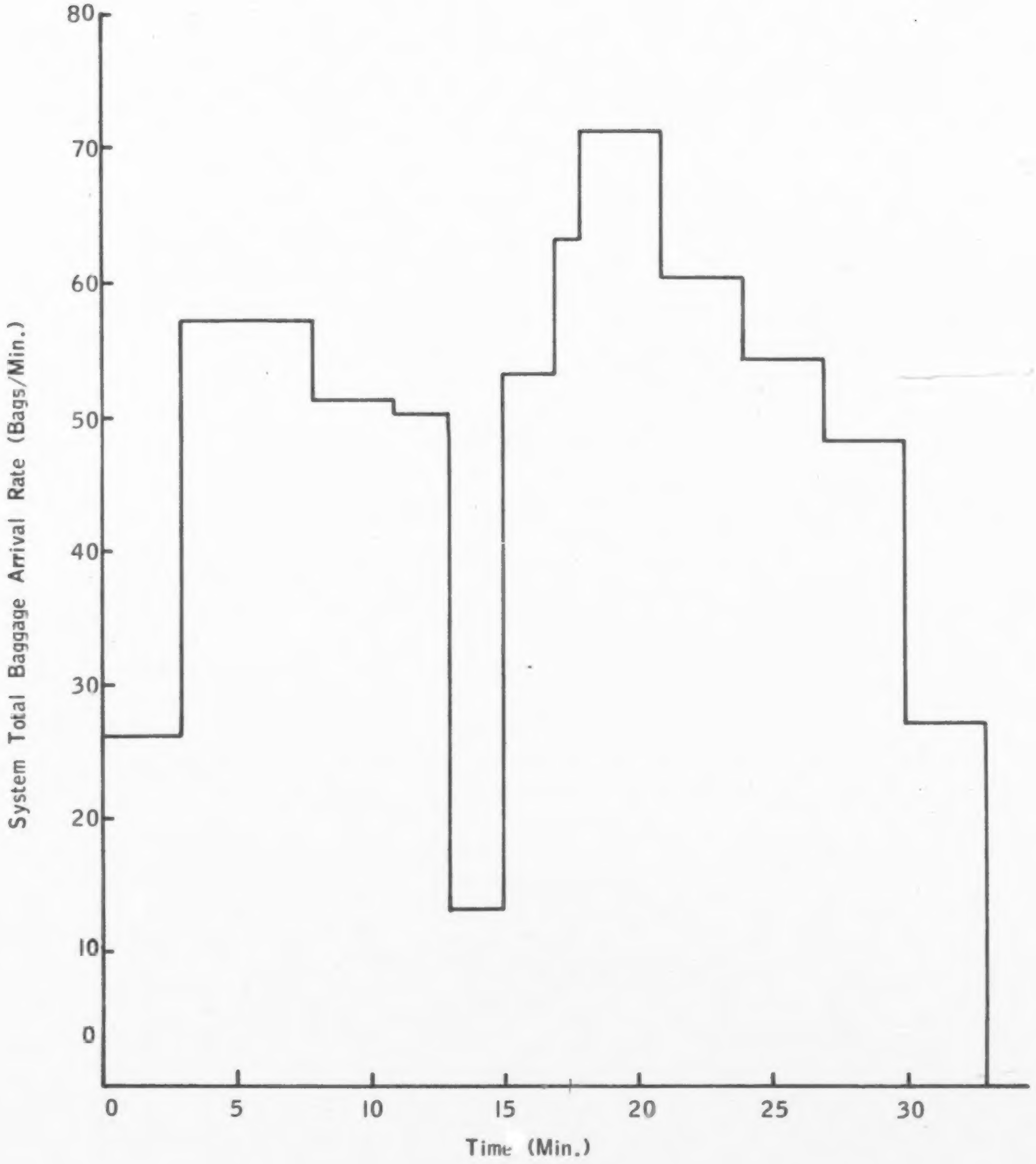
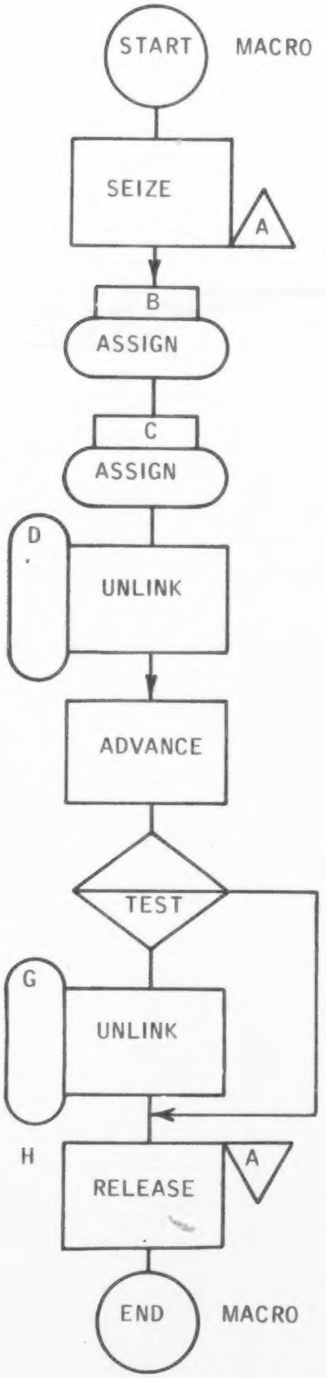


FIGURE 5
Total System Baggage Input Spectrum


START
MACRO
SEIZE
A
B
ASSIGN
C
ASSIGN
D
UNLINK
ADVANCE
TEST
G
UNLINK
H
RELEASE
A
END
MACRO

Carrier occupies the loading facility A

Encode origin B on carrier memory

Encode destination C on carrier memory

Remove one bag from bag queue D and load into carrier

Loading time elapses

Test to release another carrier for loading

Release another car to loader if necessary

Carrier leaves loader and enters the system

FIGURE 6
Model of a Carrier Loader – Load Macro

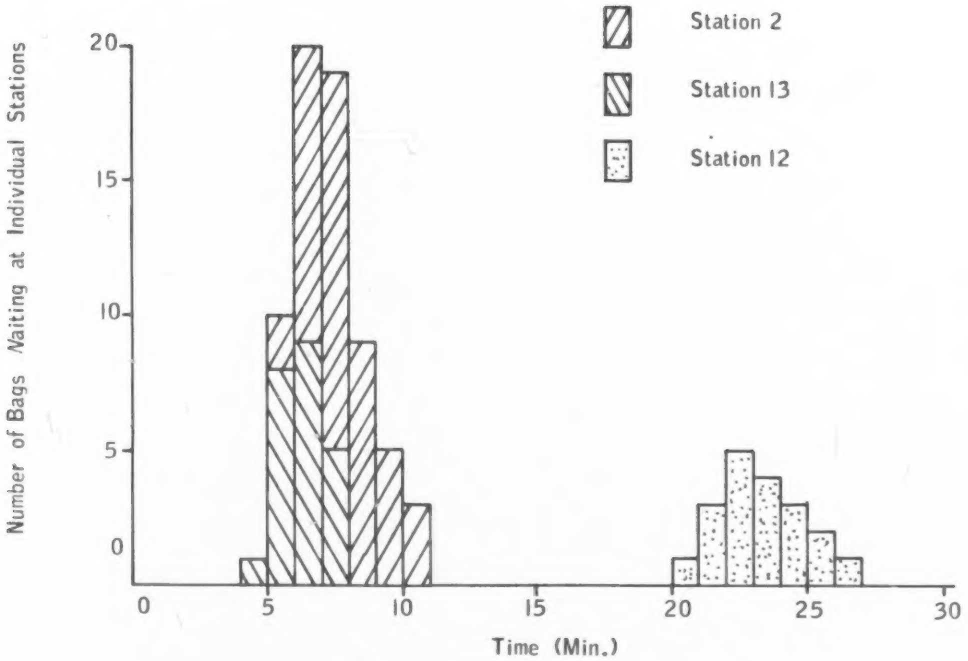


FIGURE 7
Number of Waiting Bags for
Input Spectrum of Fig. 5

```
STORAGE      CAPACITY     AVERAGE      AVERAGE      ENTRIES    AVERAGE        CURRENT        MAXIMUM
NUMBER                    CONTENTS   UTILIZATION              TIME/TRANS     CONTENTS       CONTENTS
 LN017            6         1.374         0.2291       81        30.543          2              4
 LNA02           11         5.992         0.5447      150        71.907          4             11
 LNA05            6         3.151         0.5281      148        38.842          2              6
 LNA08            7         3.310         0.4729      142        41.958          4              7
 ST030           20        15.559         0.7780       57       491.351         13             20
 ST031            7         7.000         1.0000        7      1800.000          7              7
 ST032            6         5.977         0.9961        9      1195.333          6              6
 ST033            6         5.993         0.9988        7      1541.000          6              6
 ST034           18        17.284         0.9602       25      1244.440         18             18
 ST035           28        15.596         0.5570       51       550.431         16             27
 ST036           21        11.384         0.5421       46       445.457         20             20
 ST037           24         8.805         0.3669       49       323.449         14             15
 ST040           13         9.707         0.7467       17      1027.765          8             13
 ST041            7         6.734         0.9621       18       673.444          7              7
 ST042            7         6.580         0.9400       22       538.364          7              7
 ST043            7         6.014         0.8592       40       270.650          6              7
 ST044            7         6.716         0.9594       17       711.059          7              7
 ST049            7         6.626         0.9466       20       596.350          7              7
 ST050           13        11.270         0.8669       13      1560.462          7             13
 ST051           15         1.252         0.0834        9       250.333          0              9
 ST052           33        10.412         0.3155       21       892.429          7             16
 ST053           19        13.342         0.7022       19      1264.000          3             19
 ST054           10         7.004         0.7004       10      1260.800          2             10
 ST060           30        28.869         0.9623      114       455.825         29             30
 ST061            5         1.074         0.2148       11       175.727          2              3
 ST070           40        12.016         0.3004       25       865.160          3             20
 ST071           15        14.766         0.9844       31       857.387         14             15
 ST075           33         7.265         0.2202       37       353.432          7              9
 ST076           14         6.538         0.4670       37       318.054          6              8
 ST080          261         0.000         0.0000        0         0.000          0              0
 ST081           42        42.000         1.0000       42      1800.000         42             42
 ST091          276       107.700         0.3902      136      1425.441         96            136
 ST092            7         7.000         1.0000        7      1800.000          7              7
 ST094            2         2.000         1.0000        2      1800.000          2              2
 ST095            2         2.000         1.0000        2      1800.000          2              2
 ST096            5         5.000         1.0000        5      1800.000          5              5
 ST097           14        14.000         1.0000       14      1800.000         14             14
  POR6            2         0.389         0.1944       15        46.667          1              2
  POR5            2         0.306         0.1528       11        50.000          0              1
  POR1            2         0.461         0.2303       17        48.765          0              2
  POR2            2         0.957         0.4783       35        49.200          1              2
  POR3            2         0.728         0.3639       26        50.385          0              2
```

FIGURE 8
Table of Contents of
Empty Car Storages

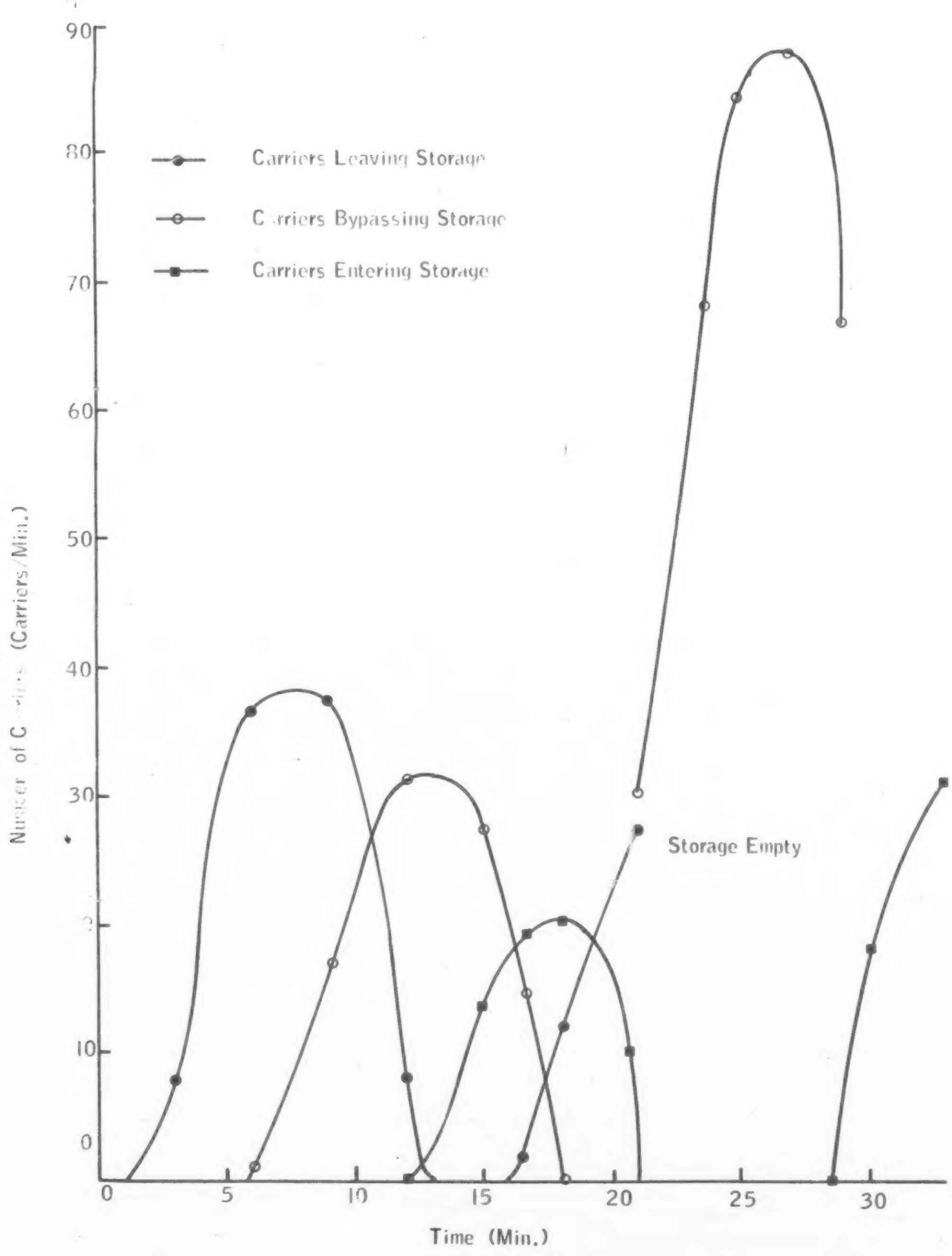


FIGURE 9
North Tunnel Traffic Activity

```
TABLE NUMBER ORB11

  ENTRIES IN TABLE        MEAN ARGUMENT      STANDARD DEVIATION        SUM OF ARGUMENTS
                24               43.750                  31.744                1050.000          NON-WEIGHTED

           UPPER        OBSERVED          PERCENT      CUMULATIVE      CUMULATIVE          MULTIPLE          DEVIATION
           LIMIT       FREQUENCY         OF TOTAL      PERCENTAGE       REMAINDER           OF MEAN          FROM MEAN
              23               9            37.50            37.5            62.5             0.526             -0.654
              25               0             0.00            37.5            62.5             0.571             -0.591
              27               0             0.00            37.5            62.5             0.617             -0.528
              29               0             0.00            37.5            62.5             0.663             -0.465
              31               0             0.00            37.5            62.5             0.709             -0.402
              33               0             0.00            37.5            62.5             0.754             -0.339
              35               0             0.00            37.5            62.5             0.800             -0.276
              37               0             0.00            37.5            62.5             0.846             -0.213
              39               0             0.00            37.5            62.5             0.891             -0.150
              41               0             0.00            37.5            62.5             0.937             -0.087
              43               0             0.00            37.5            62.5             0.983             -0.024
              45               0             0.00            37.5            62.5             1.029              0.039
              47               0             0.00            37.5            62.5             1.074              0.102
              49               4            16.67            54.2            45.8             1.120              0.165
              51               1             4.17            58.3            41.7             1.166              0.228
              53               2             8.33            66.7            33.3             1.211              0.291
              55               0             0.00            66.7            33.3             1.257              0.354
              57               0             0.00            66.7            33.3             1.303              0.417
              59               0             0.00            66.7            33.3             1.349              0.480
              61               0             0.00            66.7            33.3             1.394              0.543
              63               0             0.00            66.7            33.3             1.440              0.606
              65               0             0.00            66.7            33.3             1.486              0.669
              67               0             0.00            66.7            33.3             1.531              0.732
              69               0             0.00            66.7            33.3             1.577              0.795
              71               0             0.00            66.7            33.3             1.623              0.858
              73               0             0.00            66.7            33.3             1.669              0.921
              75               0             0.00            66.7            33.3             1.714              0.984
              77               0             0.00            66.7            33.3             1.760              1.047
              79               0             0.00            66.7            33.3             1.806              1.110
              81               8            33.33           100.0             0.0             1.851              1.173
REMAINING FREQUENCIES ARE ALL ZERO

TABLE NUMBER DEB12

  ENTRIES IN TABLE        MEAN ARGUMENT      STANDARD DEVIATION        SUM OF ARGUMENTS
                21                2.667                   5.295                  56.000          NON-WEIGHTED

           UPPER        OBSERVED          PERCENT      CUMULATIVE      CUMULATIVE          MULTIPLE          DEVIATION
           LIMIT       FREQUENCY         OF TOTAL      PERCENTAGE       REMAINDER           OF MEAN          FROM MEAN
               0              12            57.14            57.1            42.9             0.000             -0.504
               1               0             0.00            57.1            42.9             0.375             -0.315
               2               3            14.29            71.4            28.6             0.750             -0.126
               3               2             9.52            81.0            19.0             1.125              0.063
               4               2             9.52            90.5             9.5             1.500              0.252
               5               0             0.00            90.5             9.5             1.875              0.441
               6               0             0.00            90.5             9.5             2.250              0.630
      OVERFLOW, WITH AVERAGE VALUE=              18.00
```

FIGURE 10
Origin and Destination Tables

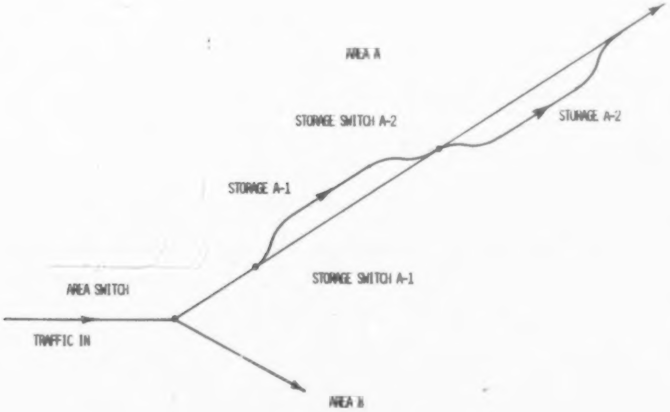


FIGURE 11
Simplified Switch Diagram

A GENERALIZED MODEL FOR SIMULATING COMMODITY MOVEMENTS BY SHIP

John C. Rea

Head, Urban and Environmental Research Division,
Pennsylvania Transportation and Traffic Safety Center, and
Assistant Professor of Civil Engineering,
The Pennsylvania State University

David C. Nowading

Assistant Professor of Transportation-Logistics
The University of Tennessee; formerly Research Assistant,
Pennsylvania Transportation and Traffic Safety Center,
The Pennsylvania State University

P. Wade Buckholts

R. Shriver Associates, Danville, New Jersey;
formerly M.S. candidate,
The Pennsylvania State University

Abstract

The paper describes the logic structure and techniques used in a Simscript program for simulating the movement of ships through a network of locks, reaches, lakes, and ports. The program also provides for endogenous route selection between "micro-route" alternatives, i.e., parallel locks or canals, and for the endogenous scheduling of ship movements, given port-to-port commodity movement demand and fleet mix. The model is basically a general network simulation tool based on the concept of a "route map" which describes the sequence of facilities to be traversed between given points in the network. The cardinal mechanism in the model is a Movement Control Module which monitors the route map of each ship during its voyage and sends the vessel to generalized satellite modules where the performance of locks, reaches, lakes, and ports are simulated. Attributes carried by the ship initialize the generalized modules to simulate a specific lock, reach, lake, or port as dictated by the ship's position along its route map. The model is being used to simulate the performance of the Great Lakes System for the Corps of Engineers.

Background

The model described in this paper was developed to assist the U.S. Army Corps of Engineers in thier assessment of the need for improvements to the Great Lakes and inland waterway systems. This is a third-generation model deriving from research sponsored by the Corps over a number of years. Initial research resulted in the models WATSIM [3] and TOWGEN [1] for simulating inland barge systems. The Corps subsequently sponsored the development of a model for simulating components of the Great Lakes System; this resulted in the Multiple Channel Deep Draft (MCDD) model [2,4,5]. During the development of the MCDD model, a number of powerful techniques were formulated which promised to form the basis of a generalized model to meet all of the Corps' needs in the area of systems simulation. A third and current research project was sponsored by the Corps to develop and apply such a model. The Network Simulation (NETSIM) model described here is the result of this research. As suggested by the acronym, NETSIM is basically a general network simulation tool; modules for simulating the operation of water navigation facilities are linked to NETSIM to provide the unique capabilities required by the Corps. To describe this specific formulation of NETSIM for the simulation of waterborne transportation systems, the acronym NETSIM/SHIP is used.

The Problem

The overall problem to which the model is addressed is that of simulating commodity movements between multiple origins and destinations by ships or barges through a network of navigation facilities. Some of the specific questions leading to the Corps' sponsorship of the research project are:

1. An appraisal of the need for a new Niagara Canal to parallel the existing Welland Canal in the light of increasing commodity movement and an evolving fleet mix
2. Determination of the response of different combined Welland-Niagara Canal configurations to imposed loading
3. Determination of the response of the

existing Eisenhower-Snell lock complex and possible new configurations to imposed loading

4. Determination of the response of the existing Sault locking system and possible new configurations to imposed loading
5. Identification of potential shipping bottlenecks in the Great Lakes System under various system states
6. The need to relate design and performance in planning future locks.

Network

The modeling problem can be disaggregated into four general areas:

1. Simulation of a transportation network--specifically, the ability to route a ship through a redundant network via a minimum or otherwise specified path
2. Endogenous assignment of ships between parallel facilities--a vessel may have to decide between parallel locks or between a series of locks and reaches, e.g., the Welland Canal versus the possible Niagara Canal
3. Endogenous scheduling of ship movements--specifically, the ability of an individual ship in the simulated system to react to ephemeral commodity movement demand and thereby schedule its next movement
4. Simulation of specific facilities--for the Corps' purposes, these are locks, reaches, lakes, and ports.

The first three problem areas are, in fact, general to many transport systems; and it is only the fourth which specifically orients the model to a shipping application. In a Personal Rapid Transit (PRT) application, for instance, the specific facilities might be switches, track segments, and stations, and the commodities to be transported would be people.

NETSIM Structure

The purpose of this paper is to present the conceptual basis of the model (Figure 1). This structure has been largely retained in implementing the model. NETSIM may be considered as consisting of three stages: the preprocessor, the simulation, and the postprocessor stages.

The Preprocessor Stage

The preprocessor stage is concerned with preparing and loading the data stream which consists of:

1. Run option and specification parameters--choice of Experience Data Bank or Event Log run; simulation run length; switches to select service look-ahead feature, parallel facilities, port rescheduling procedure, vessel file options, and input-output device options. Each of these terms will be defined in later sections.
2. Network description--the transport network is represented by nodes and links in the usual manner. Links represent transit facilities such as lakes,

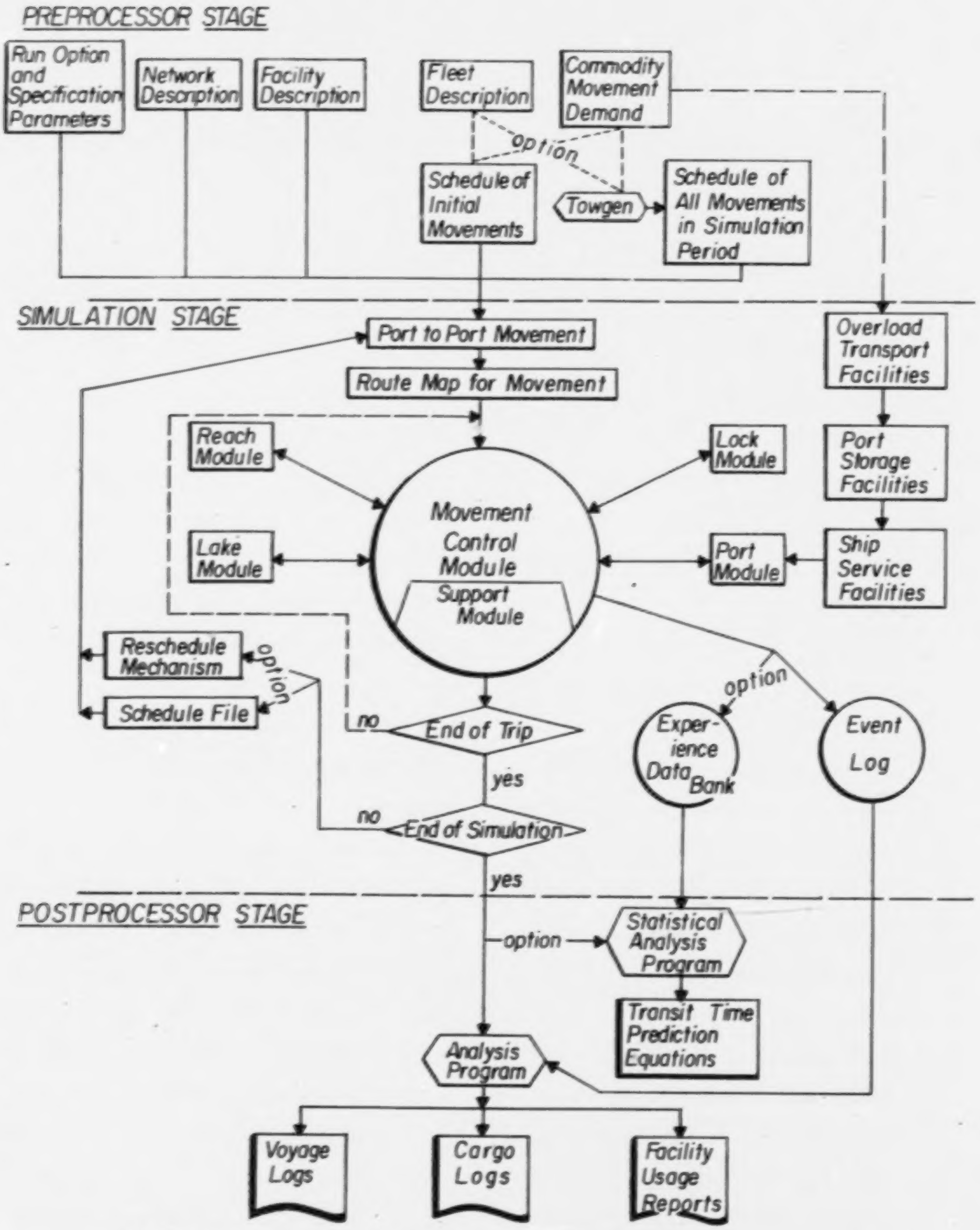

PREPROCESSOR STAGE
Run Option and Specification Parameters
Network Description
Facility Description
Fleet Description
Commodity Movement Demand
option
Schedule of Initial Movements
Towgen
Schedule of All Movements in Simulation Period
SIMULATION STAGE
Port to Port Movement
Route Map for Movement
Overload Transport Facilities
Reach Module
Lock Module
Port Storage Facilities
Movement Control Module
Support Module
Lake Module
Port Module
Ship Service Facilities
Reschedule Mechanism
option
Schedule File
no
End of Trip
yes
no
End of Simulation
yes
option
Experience Data Bank
Event Log
POSTPROCESSOR STAGE
option
Statistical Analysis Program
Transit Time Prediction Equations
Analysis Program
Voyage Logs
Cargo Logs
Facility Usage Reports

FIGURE 1 Conceptual Structure of NETSIM/SHIP.

reaches, and locks; nodes are located at facility coterminal points; and ports are also represented as nodes in the system. The actual description of the network is based on the singularity of spanning (minimum path) trees based on a given root. In effect, each port is treated as a root and heads a column which has entries for each node in the network, the column entry being the identification number of the next node in the path from the subject node to the root port. Using this formulations, one vector fully defines the route structure from any network node to a given port. A matrix of size (number of ports x number of network nodes) serves to define the route structure for the entire network.

Supplementing the above "next node" table is a facility identification matrix. Since an ordered sequence of two-node numbers defines a directional link, a matrix of size (number of network nodes squared) encompasses a mapping of directional links into navigation facility code numbers. (Clearly, there exists some redundancy here since the two tables could be combined by defining the network route structure in terms of facility identification numbers; the separation is made in NETSIM to simplify data preparation.) To allow for cases where parallel "micro-route" options exist within a macro-route link, a special code is entered in the facilities identification table. This code serves to address a parallel facilities table where the sequence of navigation facilities within each micro-route are identified.

3. System facility enumeration and description--number of ports, lakes, reaches, locks, etc. and their operational and physical characteristics.
4. Commodity movement--origin-destination quantities by type of commodity.
5. Fleet description--number, type, and characteristics of available fleet.
6. Schedule of movement--movement schedules can be specified in one of two ways. If the scheduling is to be endogenous, then only the initial origin-destination movements for each vessel are required. Subsequent movements are then determined internally. It is also possible to specify exogenously the entire schedule of movements that are to occur during the simulation period. Such a schedule can be derived by using TOWGEN [2], a model which utilizes commodity movement and fleet data to give a time-ordered list of movements. Note that an auxiliary input defining multiple-port trips for specified vessels is also possible (e.g., ore-ships on committed

shuttle movements).

The Simulation Stage

Completion of the initialization of the system by the preprocessor stage signals the beginning of execution of the simulation mechanisms embodied in the simulation stage. The approach to simulation is based on the concept of a route map which describes a vessel's current trip and is unique to each vessel. In NETSIM, a vessel carries with it four route map attributes. These are numbers representing the previous node, current node, next node, and the port of destination. Once the origin and destination of a vessel are determined, the sequence of links and nodes comprising the route from the origin to the destination port is defined by the network description "next node" array. As a vessel traverses its route, the first three route attributes are continuously updated. Note that these route attributes key into the facility identification matrix, so that the sequence of facilities (i.e., reaches, locks, etc.) comprising the route is known and that the previous facility and the next facility to be traversed are uniquely identified. When the vessel's route attributes have been updated to the point where the current node and the port of destination are identical, the current trip has been completed. It is this route map concept which enables NETSIM to deal with complex networks that incorporate alternative routing options, makes possible the use of a modular approach to the structure of the model, and facilitates the rescheduling mechanism.

Each vessel's route map is monitored by the cardinal Movement Control Module. As a vessel moves along its route, the Movement Control Module identifies three characteristics of the next link to be traversed--the type and identification number of the facility which the link represents and the direction of movement through the facility, by virtue of the node number sequence. The three characteristics are, in fact, encompassed by one attribute value. The attribute is assigned to the entity representing the vessel, and the entity is passed into the appropriate facility module where the performance simulation is effected. The entity is then passed back to the Movement Control Module where the next link in the route is identified and the processes repeated. This monitoring and referral sequence continues until the end of a ship's current trip. Note that a vessel may call at intermediate ports for a given trip en route to the final port of destination if on a committed voyage.

Linked to the Movement Control Module, in satellite fashion, are the modules which simulate the performance of the different types of facilities. In NETSIM/SHIP, these are the reach, lock, lake, and port modules. It is important to recognize that the satellite facility modules are generalized logic sequences. It is the attributes carried by a vessel as it enters the facility module which direct and enable the module to simulate the operation of a specific facility. This approach is very flexible and enables additional modules to be added easily as required to simulate the operation of any type of transportation facility. These satellite facility modules are passive until activated by a vessel's routing requirements which, in turn, are dictated by the nature of the transport system being simulated.

Common to the Movement Control Module and its satellite facility modules are many routines for searching, adjusting, referencing, and stochastic sampling. To avoid duplication, seven such routines have been assembled into a Support Module which is referenced by the other modules as necessary.

Upon completion of a given vessel's trip, the need to reschedule the vessel arises. If an exogenously specified schedule is used, another trip is triggered by the scheduled event file and the process described above repeated. If an endogenous reschedule is required, the timing and destination of a vessel's next trip is a function of the location, type, and amount of commodities awaiting shipment in the system. The character of the ship also determines its suitability for transporting the available commodities. In the absence of suitable demand at the current port, a trip must be scheduled to the nearest port at which a suitable cargo is available. Since ship rescheduling is intimately associated with the port simulation module, further discussion is delayed until the logic of the port module is outlined.

The Postprocessor Stage

Upon completion of the simulation period, the third, or postprocessor, stage of NETSIM comes into effect. The function of the postprocessor is to generate system performance reports from the coded event file that is the output of the simulation stage. The event file lists in time-sequence all events which occurred in the simulated system during the simulation period. These recorded events are subsequently analyzed using a simple Fortran program to produce a set of statistical reports. This approach was adopted to minimize the time required for the actual simulation on a large computer and to provide maximum analytical flexibility. In addition, this approach enables each potential user to produce reports suitable to their own needs. It is a relatively simple matter to make appropriate changes to the existing NETSIM/SHIP postprocessor program to augment its report generation capability. The event file approach offers another advantage in that the simulation need not be rerun to obtain supplemental performance reports; it is only necessary to rerum the taped event file through additional postprocessor programs.

Ship Navigation Modules

The place of the reach, lake, lock, and port simulation modules in the model structure was described in the last section. Some of the features of these modules are now described.

Reach Module

The reach module represents ship transit time by sampling from a transit time probability function appropriate to the particular reach in question. The function may be derived from empirical data, or it may be a theoretical function. The module incorporates an optional reach-specific no passing rule which allows a trailing vessel to overtake but not to pass a preceding vessel in a reach.

Lake Module

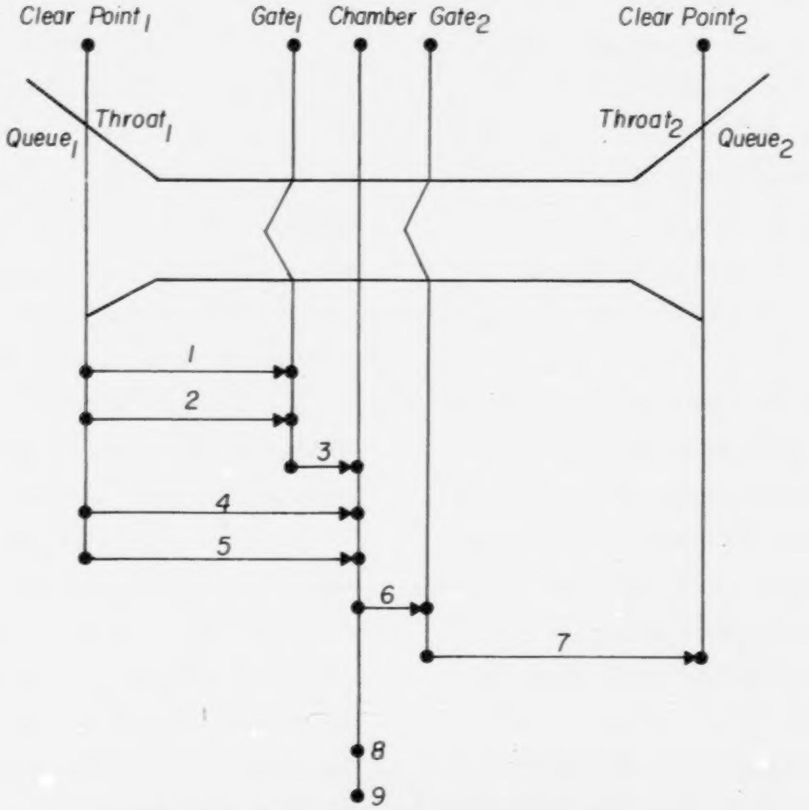
The lake module functions in a fashion similar to the reach module except that no constraints are imposed on passing. An internodal distance matrix is specified for each lake in the input stream. When a ship is to cross a lake between given node points, this matrix is referenced to obtain the appropriate distance. Lake transit times are derived by sampling from a standard cumulative density function (CDF) which is adjusted according to the distance to be traversed on the lake and the characteristics of the subject vessel.

Lock Module

The lock module has the most complex logic structure. This intricacy results from the inherently complex nature of an efficient locking operation and was dictated, in part, by the need to simulate the performance of a lock so that it is sensitive to engineering design features. Figure 2 depicts the nine time elements in relation to the physical lock configuration that are used to simulate lock operations. Five time elements are used for entry maneuvers, two for the exit maneuvers, one for chamber processing (to change water level with a vessel in the chamber), and one for recycling the lock (to change water level in an empty chamber). The entry and exit CDFs are direction differentiated and can be adjusted for differences in vessel performance.

Two service rules are included in the logic structure of the lock module, their selection depending upon conditions at the lock. When queues exist on both sides of a lock, a Serve-Opposing-Queues-Alternately (SOQA) service rule is adopted. If a queue exists only on one side of a lock, then a First-Come-First-Served (FCFS) service rule is adopted. These are standard operating procedures for locks operated by the St. Lawrence Seaway Authority and for locks under the control of the St. Lawrence Development Authority.

These service rules alone do not, however, suffice to simulate lock operations in the manner adopted by experienced lockmasters. For example, on the Great Lakes System lockmasters have radio communication with approaching vessels so that they can anticipate vessel arrivals and can make appropriate operational decisions. Such decisions are replicated by means of two "look ahead" features in the logic structure. The first of these is the "service look ahead" mechanism. Prior to making a decision to recycle a lock to accommodate a waiting or approaching vessel, the service look ahead scans the adjacent reach on the opposite side of the lock for approaching vessels. If a vessel in this reach


Clear Point$_1$
Gate$_1$
Chamber
Gate$_2$
Clear Point$_2$
Throat$_1$
Queue$_1$
Throat$_2$
Queue$_2$
1
2
3
4
5
6
7
8
9

1. Clear point to short-entry position, moving start.
2. Clear point to short-entry position, stationary start.
3. Short-entry position to chamber.
4. Clear point to chamber, moving start.
5. Clear point to chamber, stationary start.
6. Chamber to gates-clear point.
7. Gates-clear point to clear point.
8. Process (with vessel).
9. Recycle (empty).

FIGURE 2 Schematic of Lock Time Elements used in NETSIM/SHIP.

could enter the lock chamber at its current water level before the opposing vessel could enter the recycled lock, the lock recycle is suppressed. The second feature is the "recycle look ahead" feature. In the absence of opposing traffic, the recycle look ahead adjusts the water level in the lock chamber to receive an approaching vessel directly into the chamber. If the vessel arrives before the recycling is complete, the vessel waits in the short-entry position until entry is possible.

Port Module

The immediate applications of NETSIM/SHIP envisaged by the Corps of Engineers do not require an elaborate port module. The current model, therefore, determines ship turnaround time simply by sampling from port-specific CDFs. The mechanisms to support a more elaborate module are, however, built into the structure of NETSIM/SHIP.

The logic structure of an elaborated port module has been defined with a port considered to have attributes relating to berthing capacity, ship servicing capacity (e.g., craneage), commodity-specific storage capacity, and seasonal attributes which define the opening and closing dates of the port. Each port also maintains an incoming-ship list that contains the identification of every ship that currently considers the subject port as its next port-of-call. The time spent in port by a ship is a function of berth and servicing availability and the amount of cargo to be on- and off-loaded. The commodities in a port awaiting shipment are a function of port storage capacity, inputs from the overland transportation system, and previous cargo movements from the port.

After on- or off-loading cargo, a ship must be rescheduled (under the endogenous scheduling option) and may be in one of three states. If the vessel is on a committed voyage, it will have a predefined next port-of-call, the current port being an intermediate stop. In this case, a trip to the next port-of-call is scheduled, with cargo if cargo is available or in ballast if not. If the vessel is not on a committed run, a completely new voyage must be scheduled. If suitable cargo exists in the current port, the next voyage is scheduled to accommodate this commodity movement. If no such cargo exists, a search for suitable cargo at other ports must be instituted, starting at the nearest one. To obviate sterile in-ballast trips, the incoming ship list of each port must be checked and the available cargo manifests at that port adjusted to account for commodity movements which will occur before the subject vessel can reach that port. When the location of suitable cargo is identified, a voyage to the nearest such port is scheduled for the subject vessel.

Assignment Decision Technique

In ship navigation contexts (excluding oceans), redundant networks rarely exist in the sense that alternative, nearly competitive, routes are not usually available. However, within a macro-route, alternative micro-routings may exist.

Examples are twinned locks or parallel canals, the Welland-Niagara canals being a case in point. This situation is common in many transport systems.

Decisions as to which macro-route to use are usually easily made by observation or, if need be, by a minimum path algorithm. Selecting between alternatives at the micro-scale within a macro-route usually depends upon the conditions prevailing in the micro-route alternatives. The NETSIM assignment decision technique is based on this philosophy. It is assumed that it is possible to derive an equation to relate expected transit time through a sequence of facilities to the traffic conditions prevailing in those facilities. The mechanisms to support the derivation of these equations is built into the NETSIM/SHIP logic. Prior to simulating the operation of a system containing micro-route alternatives, it is necessary to derive a set of expected transit time prediction equations for each such alternative.

The equations are derived by regression analysis of an Experience Data Bank (EDB) which is built up by simulating the operation of each micro-route alternative individually. The approach is shown graphically in Figure 3. In a simulation run to construct an EDB, as a ship passes the assignment decision point, a snapshot is taken of current traffic conditions in the subject micro-routing alternative; as the ship leaves, the actual transit time is recorded. These observations are made for each ship, differentiated by direction, and constitute the EDB.

A set of dummy transit time predictor equations of the form $E(T) = C_o + \sum_{i=1}^{n} C_i X_i$ are already built into NETSIM/SHIP. As a result of the EDB analysis, the user simply calibrates these equations by specifying the influencing variables (X_i) and assigning coefficient values (C_i).

When running a system simulation with calibrated equations, values of the appropriate traffic condition variables are automatically obtained for each micro-routing alternative when a ship reaches the appropriate assignment decision point. The expected transit time is computed for each alternative using the transit time prediction equations, and the ship is assigned to the micro-route offering the least expected transit time.

Language

Although the earlier MCDD model was constructed using IBM's General Purpose Simulation System (GPSS), it was decided to use Simscript for programming NETSIM/SHIP. GPSS was selected for the MCDD model because it could encompass the restricted objectives of the MCDD model and it offered considerable savings in programming effort. In retrospect, the selection of GPSS was correct for that purpose since it allowed the MCDD model to be programmed within severe time constraints.

While the MCDD model development had limited objectives, the generality and power of

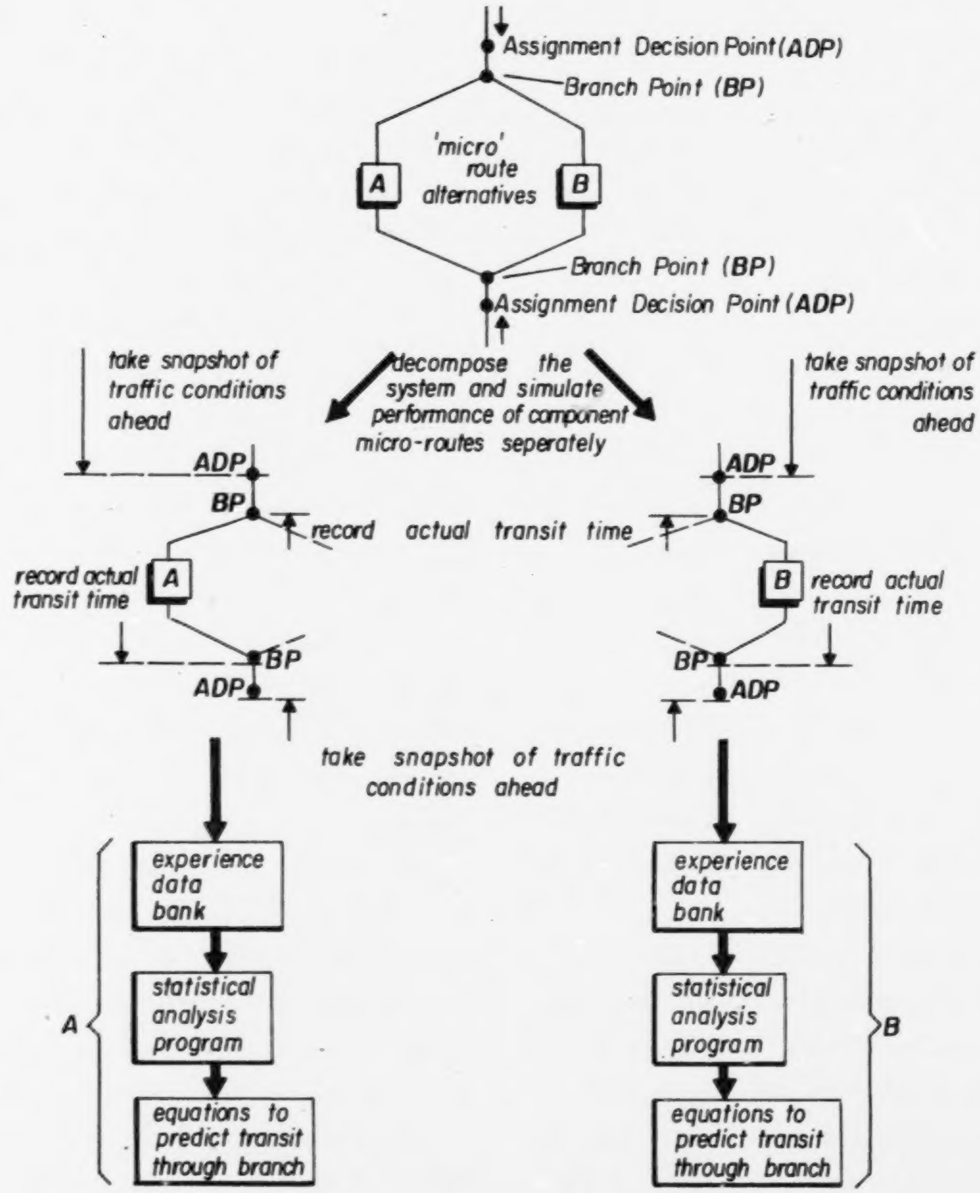

Assignment Decision Point (ADP)
Branch Point (BP)
'micro' route alternatives
A
B
Branch Point (BP)
Assignment Decision Point (ADP)
take snapshot of traffic conditions ahead
decompose the system and simulate performance of component micro-routes seperately
take snapshot of traffic conditions ahead
ADP
BP
record actual transit time
ADP
BP
record actual transit time
A
B
record actual transit time
BP
ADP
BP
ADP
take snapshot of traffic conditions ahead
experience data bank
statistical analysis program
equations to predict transit through branch
A
experience data bank
statistical analysis program
equations to predict transit through branch
B

FIGURE 3 Derivation of Assignment Decision Equations.

the NETSIM concept made it impossible to prespecify its potential uses and applications. Certainly, even in a shipping context, it is easy to envisage model capabilities beyond those actually required by the Corps of Engineers for the current research project. Simscript, being a general purpose language, was selected since it offered the inherent ability to encompass easily any developments of NETSIM. In addition, the English-like readability of the language allows a Simscript written program to virtually serve as its own documentation. This is a distinct advantage in a complex model.

In contrast to the four man-months required to program the MCDD model in GPSS, the programming of NETSIM/SHIP in Simscript has been much more protracted. A detailed model specification for NETSIM/SHIP, including logic structures and techniques, resulted from a review of the MCDD model. Some nine man-months of Simscript programming effort and four thousand dollars worth of computer time on an IBM 360/67 was subsequently required to bring NETSIM/SHIP to its present state of development.

Future Extensions of NETSIM

The completion and documentation of the NETSIM/SHIP capabilities described in this paper is currently in hand at The Pennsylvania State University's Transportation and Traffic Safety Center, and the model is being applied to the Great Lakes-St. Lawrence Seaway system. The logic structure for facility simulation modules to allow NETSIM to be applied to networks oriented to personal rapid transit, highway, traffic light, and airport systems is being considered.

References

1. Bronzini, Michael S., Waterway Systems Simulation: Volume III--TOWGEN: A Tow Generation Model for Inland Waterway Simulation. Report TTSC 7110, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.
2. Carroll, Joseph L., and Bronzini, Michael S. Waterway Systems Simulation: Volume I--Summary Report. Report TTSC 7108, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.
3. Gimbel, John H., III, Waterway Systems Simulation: Volume II--WATSIM: A Waterway Transport Simulator. Report TTSC 7109, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.
4. Rea, John C., and Nowading, David C. Waterway Systems Simulation: Volume V--Simulation of Multiple Channel Deep Draft Navigation Systems. Report TTSC 7112, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.
5. Rea, John C., and Nowading, David C. A Simulation Model for the Study of Two-way Ship Traffic Through Canals Which Offer Multiple Routing Options. Technical Note 55, Pennsylvania Transportation and Traffic Safety Center, The Pennsylvania State University, University Park, Pennsylvania, 1971.

Key Words

SIMULATION, SIMSCRIPT, SHIPPING, LOCKS, GREAT LAKES, PORTS, COMMODITY MOVEMENTS, NETWORK SIMULATION

SIMULATION OF GARLAND, TEXAS, VEHICULAR TRAFFIC USING CURRENT AND COMPUTED OPTIMAL TRAFFIC SETTINGS

Frank P. Testa
Mark Handelman

Surface Transportation Systems Department
Federal Systems Division
INTERNATIONAL BUSINESS MACHINES CORPORATION

Abstract

This paper presents results of a study utilizing computer simulation of vehicular traffic in the downtown area of Garland, Texas. A general discrete digital simulation model, the Vehicle Traffic Simulator (VETRAS), developed by IBM Corporation, was used for the simulation. Using data supplied by the City of Garland, traffic patterns for three peak periods of daily operation–A.M., Noon, P.M.–were simulated. Two simulations were run for each period. In the first, intersections were controlled with signal settings currently in use in Garland. In the second, intersections were controlled by signal settings derived via a pattern optimization algorithm. A minimum interference technique was used to compute coordinated signal settings and offsets to maximize arterial and network performance.

The results show that arterial and network performance improves in each of the peak periods using the computed signal settings. Further, there is a direct relationship between volume and relative improvement. Estimated cost benefits for these improvements are also presented.

THE PROBLEM

The city of Garland, Texas is considering the implementation of a real time computer system for control of its vehicular traffic. Traffic signal patterns corresponding to peak periods, weekends, special events, etc., would be generated, using a traffic responsive optimization technique. The system would select and apply the appropriate pattern for a given situation in response to actual traffic demand. Data gathered by the system from online traffic detectors would be used to update the patterns. In addition, the system would monitor and report traffic network performance.

A means was sought to quantify the expected improvements from such a system since actual installation of the necessary hardware and software, even for a limited trial area, would be expensive. It was decided to use digital simulation for evaluation of the improvements possible from the proposed system. The simulation would provide the data necessary for a comparison of the performance of Garland's current traffic signal settings with settings representative of optimized patterns that would be used for computer control during AM, NOON, and PM peak periods.

THE MODEL

The simulations were performed using IBM's Vehicle Traffic Simulator (VETRAS), a general purpose discrete simulation model. VETRAS is written in IBM's General Purpose System Simulator language, GPSS/360, chosen for its ease of programming and timekeeping and statistics gathering features.

VETRAS simulates vehicle traffic moving through a network of streets and intersections. It is designed to be an aid in analysis of traffic control techniques. Some of the statistics gathered by the model are:

a. Average time cars spent in queues

b. Trip times

c. Percentage of cars that did not have to stop for traffic lights

d. Lane utilization.

The user can specify network, vehicle, and control parameters. The main elements of the VETRAS structure are:

a. Geometry

b. Signal control

c. Vehicles.

GEOMETRY

The network geometry consists of lanes and intersections into which the lanes empty. Lanes are grouped into segments composed of adjacent lanes carrying traffic in one direction between two intersections. Intersections are the regions common to two or more intersecting lanes where there is usually some competition for the right of way. The intersection of any two lanes determines a cell. Thus, each intersection is divided into a number of cells equal to the product of the number of intersecting lanes. Routes through the intersection for each approach lane are given as sequences of cells.

All lanes, segments intersections, and cells used to describe a network must be uniquely numbered. Figure 1 shows a sample network and Figure 2 shows a sample intersection.

CONTROL

The movement of vehicles into an intersection is controlled by signal light phases, with one or more phases controlling traffic streams that have simultaneous right of way. VETRAS permits two types of phase control for an intersection–fixed and acutated. A fixed phase is defined by a cycle length, split and offset:

a. Cycle length is the total time for a single sequence of red and green.

b. Split is the percentage of the cycle given to green time.

c. Offset is a percentage of the cycle used for initial synchronization of related phases.

Intersection 1 in Figure 1 is a simple two-phase intersection. Phase one controls the East/West lanes, while phase two controls the North/South lanes.

For intersections under actuated control, one or two sequences of phases (called step sequences) are used. A phase regulator is generated for each sequence. It steps through each phase in turn, setting it green and the others red. The amount of time a given phase remains green may vary, depending on traffic demand.

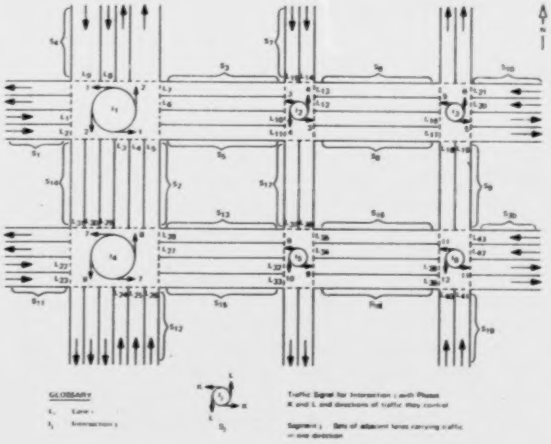


Figure 1. VETRAS Network Geometry

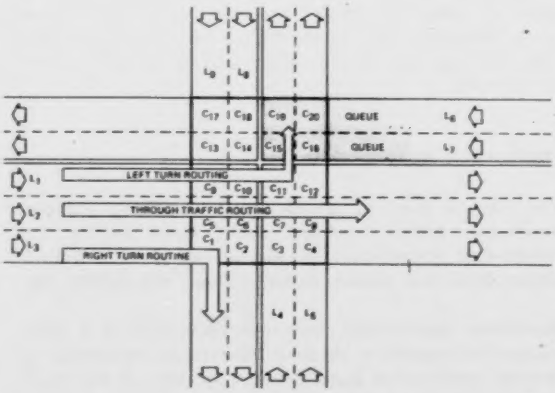


Figure 2. VETRAS Intersection Geometry

Actuated phases are of three types:

a. Main fixed time off

b. Main demand off

c. True actuated.

Main type phases will always turn green when encountered in a step sequence. Fixed time off phases will remain green for a constant time, then turn red. Demand off phases will be green for some minimum time and then remain green until a demand is presented at an intersection access controlled by another phase.

A true actuated phase will turn green when encountered in the step sequence only if a vehicle demand is sensed, and it will stay green some minimum time. If additional demand is detected within some given detect interval at the end of the minimum green time, the phase will remain green for some given additional time. This process is repeated up to some maximum allowable green time.

VEHICLES

Vehicles are input to the network on peripheral lanes. In Figure 1, Lanes 1, 2, 8, 9, 14, 15, etc. represent possible input lanes. Special internal sources and sinks of traffic, such as parking lots, can also be introduced by specifying them as additional input and output locations. For each specified input lane a mean time between arrivals, t_a, and a standard deviation, σ_a, must be supplied. Vehicles are generated at the lane entry points every $t_a + k\sigma_a$, where k is a random variable such that $-.999 \leq k \leq +.999$. When a vehicle is generated, a number of operating characteristics are assigned by the model. These include length, speed, intervehicle gap, and route through the intersection ahead. Routes are assigned on a percentage basis where the percentages of right and left turns are input for each segment.

Each vehicle moves down the lane until it reaches either an intersection or a queue of other vehicles. Vehicles in a queue move up toward an intersection until they are first in the queue, whereupon the vehicle will move into the intersection only if the appropriate phase is green. Figure 3 is an overview of VETRAS.

SIMULATED NETWORK

The city of Garland supplied data describing network geometry, traffic flow and traffic signal settings for twenty-four intersections in the central business district during three peak periods of traffic flow - AM, NOON, PM.

Bandwidth optimization techniques described in a later section were applied to the flow data provided by Garland to develop synchronous signal settings for each of the three peak periods. The signal settings supplied by Garland will be called the current signal settings, and the bandwidth optimization-derived settings the computed signal settings. The latter are of the type used for computer control and are designed to maximize the flow of a traffic network in response to traffic demands.

Two simulations for each of the peak periods were conducted. Current signal settings were used to control traffic in the first simulation while computed signal settings were used in the second. The traffic flow rates input to each pair of peak periods simulations were identical and were derived from the data supplied by the city of Garland.

The portion of Garland, Texas included in this simulation has the following inclusive boundaries:

a. East–First Street

b. West–Garland North Star

c. North–Walnut Street

d. South–Avenue D.

The major arteries in the network included the border streets mentioned above, as well as the following:

a. North-South–Glenbrook Drive and Fifth Street

b. East-West–State Street and West Garland Avenue.

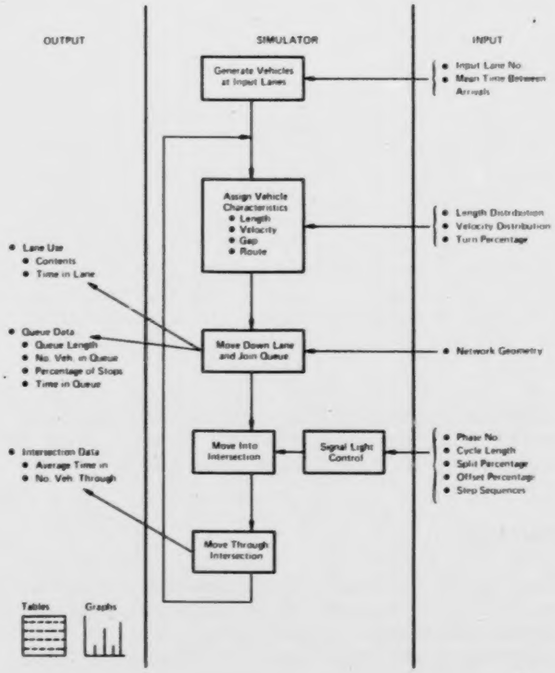

OUTPUT
SIMULATOR
INPUT
Generate Vehicles at Input Lanes
• Input Lane No
• Mean Time Between Arrivals
Assign Vehicle Characteristics
• Length
• Velocity
• Gap
• Route
• Length Distribution
• Velocity Distribution
• Turn Percentage
• Lane Use
• Contents
• Time in Lane
• Queue Data
• Queue Length
• No. Veh. in Queue
• Percentage of Stops
• Time in Queue
Move Down Lane and Join Queue
• Network Geometry
• Intersection Data
• Average Time in
• No. Veh. Through
Move Into Intersection
Signal Light Control
• Phase No
• Cycle Length
• Split Percentage
• Offset Percentage
• Step Sequences
Move Through Intersection
Tables
Graphs

Figure 3. VETRAS Overview

600

The network is shown in Figure 4. There are 24 signalized intersections currently controlled either by fixed cycle phases, traffic actuated phases, or flashing signals. A single control policy is in force for the entire day, and no arterial progression scheme is currently in use.

For intersections currently under actuated control, phase time given to an arterial approach may be distributed among straight and turn phases. The turn phases are actuated and will not be given green if there is no demand. For straight phases, the A flow is a main demand off type phase, and the B flow is a main fixed time off phase. These A and B phase type assignments also hold for intersections under two-phase actuated control.

In the simulations using computed signal sets, all main type phases are fixed time off. Any green time not used from the maximum allocated green time of an associated actuated turn phase is given to the main phase during any cycle. In this fashion, total green time allocated to a given direction will be used according to the demand. However, the green time will always terminate after some fixed time interval to preserve the computed offset relationships among phases on an artery.

In addition to the signalized intersections, "dummy" intersections have been included in the network. Dummy intersections have no signal control and serve several purposes in the simulation:

a. Realistic modeling of left turn space

b. Model non-signalized intersections

c. Provide for traffic gains and losses along an artery due to non-modeled intersections.

DERIVATION OF COMPUTED SIGNAL SETTINGS

The current method used for signal control in Garland is based upon one set of fixed time and vehicle-actuated control settings. This method of signal control is not responsive to changing network states. Furthermore, there are no synchronized traffic signals along heavily traveled arteries. As a consequence, when this control discipline was imposed on the simulation of the Garland traffic network, large queues were seen to form at intersections along heavily traveled arteries during peak traffic periods.

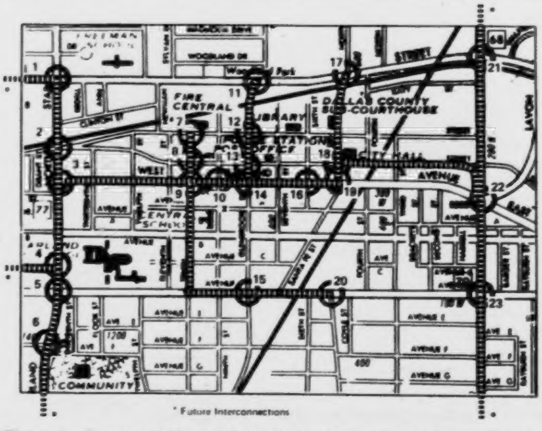
To demonstrate possible improvements, a traffic pattern optimization algorithm was used to develop new control strategies. This algorithm is based on the concept of adaptive computer control of traffic signal settings. It provides for optimization of traffic flows within a network based on maximized bandwidth.

Figure 5 shows a time-space diagram for traffic movement along an artery. The horizontal line segments indicate the red times at each intersection and the gaps between them correspond to green times for each signal cycle. The sloping bands represent the bandwidth up and down the artery for a given velocity. A vehicle whose travel trajectory is confined to one of these bands can travel unimpeded the length of the artery in the direction of the band.

The procedure used for bandwidth optimization along an artery can be briefly described as follows: given a base cycle and green times for each intersection, the algorithm determines a maximum bandwidth for a given range of velocities and computes the offsets necessary to coordinate the signals.

In this study, the maximum through-band for each artery was computed by the above procedure for a fixed cycle length of fifty seconds. This was representative of currently used cycle lengths, and analysis of traffic flow data supported its use. The allocation of green splits for all phases of each intersection was computed proportional to directional traffic flows.

The computations for each artery were performed over a velocity range of 30 - 50 mph. The intersections that were included for each artery were selected based on a criteria of major flow contributions in both a North-South or East-West direction. Those intersections which did not provide linkage between major crossing arteries were not used in the bandwidth calculation. Consequently, these intersections



* Future Interconnections

Figure 4. Portion of Map of Garland Showing Network Simulated

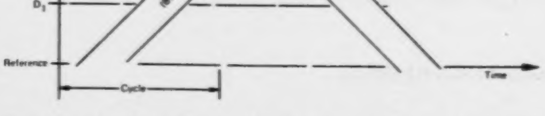


Figure 5. Space-Time Diagram Showing Bandwidth for Arterial Progression in Two Directions

were included as flashers in the traffic simulation for both current and computed signal settings.

Once the maximum bandwidths were computed for each artery, a performance index was calculated to determine the best combination of a North-South or East-West artery and all opposing arteries for optimization of network flows.

The arteries chosen for optimization for each of the peak flow periods were as follows:

a. AM – Walnut Street and all North-South

b. Noon – Garland/North Star and all East-West

c. PM – Garland/North Star and all East-West.

Once the arteries for optimization were selected, the offsets computed via the bandwidth calculation were synchronized on a network basis by selecting a reference intersection and adjusting each offset relative to that intersection.

OPERATIONAL CONSIDERATIONS

Data supplied by Garland and computed signal settings were translated to punched cards in a form acceptable to VETRAS and put in a data base on a direct access storage device. VETRAS itself, and a GPSS Output Edit report generator were also resident on direct access data sets. The simulation output was written to a direct access device, later archived to tape. The report generator was used with the simulation output as its data base to produce output reports for analysis and inclusion in the report made to the City of Garland.

No modifications to the VETRAS code were necessary. The majority of presimulation effort was in preparing the Garland data–geometry, traffic, signal control–for input to VETRAS and building the data base. The analysis, input formatting, and the construction and checking of the data base required approximately 1 month.

A total of 10 simulation runs were conducted. The first four simulation runs did not contain signal control. One run was made to check the Garland geometry as described to VETRAS, and three runs were made to check the AM, Noon, and PM peak traffic patterns. Two runs of 15 minutes simulation time were then conducted for each peak period (one with the current signal settings and another with the computed signal settings).

All simulations were run on an IBM System/360 Model 65, using one 2314 disk pack for direct access storage. Simulation of 15 minutes of traffic time required an average of 8 minutes central processor, or CPU time. Figure 6 gives an overall view of the operational process.

SIMULATION RESULTS

Flow volumes into the network and turn percentages at each intersection corresponding to each of the three peak periods, as supplied by Garland, were input. For each peak period, runs were made with the current signal settings and with the computed signal settings derived for that peak period. Simulation runs of 15 minutes were considered sufficient, as data sampling showed the model stabilizing. This is reasonable since the distributions of trip times show mean times of 2 minutes or less.

The histograms of Appendix A show comparisons of certain system performance criteria. For each peak period, distributions of queue waits, queue lengths, and trip times are graphed. The light bars give the distributions under current signal sets, while the dark bars represent distributions under computed signal sets.

Figure 7 shows an example of arterial flow performance comparisons that were plotted for each of the three peak periods. The results of 15 minutes of simulation under current signal settings are plotted against the results of 15 minutes of simulation under computed signal settings. Four quantities are plotted for each artery. They are:

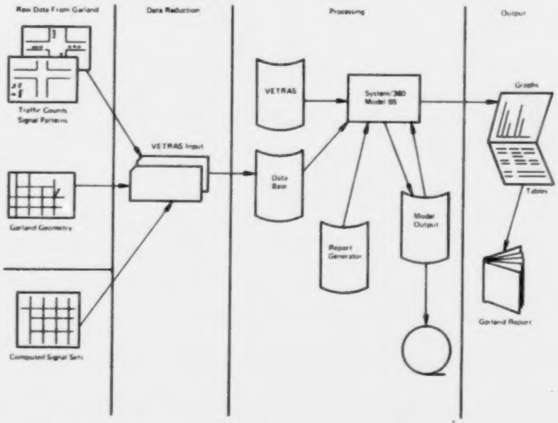


Figure 6. Data Reduction and Simulation Process Overview

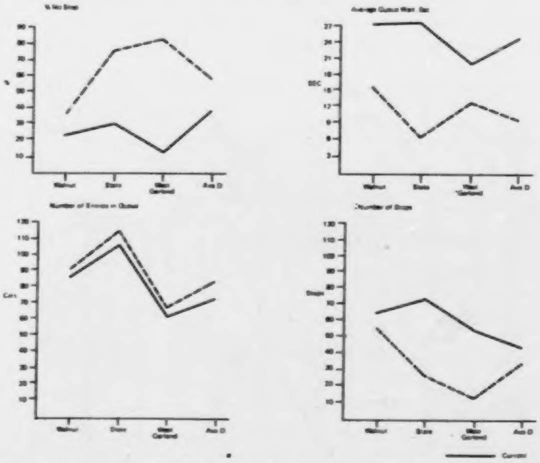


Figure 7. AM Peak, Garland-Northstar Traveling South, 15 Minutes Simulation

a. Percent Queue Zeros–The percent of cars that did not have to stop before entering an intersection.

b. Average Queue Wait–The average time, in seconds, that stopped cars had to wait before entering an intersection.

c. Number of Entries in Queue–The number of cars that have approached an intersection.

d. Number of Stops–The number of cars that had to stop before entering an intersection.

A study of similar statistics gathered for all arteries indicates queue waits and number of stops are generally lower along an artery under the computed signal sets than under current signal sets. Throughput, as measured by number of entries in the queue, is generally somewhat higher under computed signal sets. The percent of queue zeroes is also higher for computed signal sets.

As might be expected, the computed signal sets also resulted in better overall network performance. Table 1 shows a comparison of several general network statistics taken after 15 minutes of simulation. Note that two queue waits are given. The average queue wait includes cars that did not stop before entering an intersection. The average queue wait includes only cars that had to stop.

A direct relationship between traffic volume and relative improvement in network performance can be seen in Table 1. For the NOON, AM, and PM peaks, which have successively heavier flows, the improvements in each of the measured quantities are successively greater.

The improvements discussed above result from application of the bandwidth optimization algorithm to compute coherent arterial offsets, cycle lengths, and splits as a function of demand. Improvements over current control would also be possible using non-optimizing procedures to compute arterial offsets for current cycle lengths and splits. Altering cycle lengths and splits as a function of time of day would also bring improvements over current performance.

The greatest improvements are, however, expected by using a computer in a traffic responsive mode. The bandwidth optimization algorithm is used to compute offsets and signal sets based on demand, where the demand is automatically sensed and reported to the monitoring computer. The demands are used to update the signal patterns and to select appropriate patterns for use when needed. The demand-monitoring, signal pattern optimization and pattern selection functions performed by the control computer provide a responsive and flexible control system.

Expected improvements for Garland compare favorably with the results of similar analyses conducted for other cities as shown in Table 2.

COST ANALYSIS

The cost to the motoring public during each of the peak periods under control of the current and computed signal settings can be estimated. Figures for the cost computations are taken from the American Association of State Highway Officials report entitled Road User Benefit Analysis for Highway Improvement. This report determines the cost for stopping a vehicle from various speeds, plus the cost of a standing delay.

Table 1. Network Performance Comparison

Item	AM		Noon		PM		Improvement (%)		
	Current	Computed	Current	Computed	Current	Computed	AM	Noon	PM
Throughput (Number of Cars)	1298	1354	1146	1180	1366	1523	4.3	3.0	11.5
Queue Zeros (%)	58.9	66.9	65.8	67.3	57.0	70.4	13.6	2.0	23.0
Average Queue Wait (Seconds)	9.0	5.5	6.9	4.8	11.9	4.9	39.0	33.0	63.6
$Average Queue Wait (Seconds)	21.9	16.6	20.2	14.7	27.7	16.6	24.0	27.0	40.0
Average Queue Length (Number of Cars)	2.3	1.9	1.8	1.6	2.75	2.15	17.0	11.0	22.0
Average Trip Time (Minutes)	2.0	1.6	1.6	1.3	2.2	1.6	20.0	18.5	27.0
Number of Stops	2940	2433	1832	1744	3456	2473	17.0	5.0	28.0

Table 2. Improvements Comparison

Item	Garland	San Jose	Wichita Falls
Intersections	26	32	80
Estimated Savings/year ($)	118,830	250,000	4,200,000
Stop Probability Reduction (%)	17	17.8	8

The elements of cost used in the report are based on national averages and are delineated as follows:

Gasoline = $0.32 per gallon

Oil = $0.45 per quart

Tires = $100 per set initial cost

Time = $1.55 per hour.

For a speed of 30 miles per hour, which is the posted speed in the network simulated, the following figures are given:

0.74 ¢ = Cost of a vehicle stop

0.008¢ = Cost per second of idling

0.043¢ = Cost per second of waiting.

Using these figures, a formula for cost per stop is:

$$0.74¢ + \text{stop time } (0.008 + 0.043) = ¢ \text{ per stop}$$

where stop time is in seconds. The average queue wait for stopped cars can be used for this figure.

Table 3 presents a summary of estimated costs and savings for each of the peak periods. The formulas used are:

Cost/hour

$$\frac{\text{Cost/stop x Number of stops/15 min x 4}}{100}$$

$$= \$ \text{ Cost/hour}$$

Saving/year

$$\text{Saving/hour x 2 x 260} = \$ \text{ Saving/year}$$

In the cost per hour formula, the number of stops observed in 15 minutes of simulation is extrapolated linearly to obtain number of stops per hour by multiplying by four. In the saving per year formula a 2-hour peak period is assumed and a 260-day work year (365 - 2x52) is used.

For each of the peak periods, the average time per stop is less under the computed signal sets than under the current signal sets. This results in a lower cost per stop. This difference is most noticeable in the PM peak figures, where the cost per stop is 26 percent less under computed signal settings. In addition, the number of stops during each of the peak periods is less under the computed signal sets. This, combined with the lower cost per stop produces a lower cost per hour. For the PM peak period, there are 28 percent fewer stops under computed settings. There is, however, a 48 percent difference in the cost per hour. Table 4 summarizes the differences in costs for each of the peak periods.

Table 3. Cost Comparison of Current and Computed Signal Settings

Item	AM		Noon		PM	
	Current	Computed	Current	Computed	Current	Computed
Stops per 15 min	2940	2433	1832	1744	3456	2473
Average Queue Wait (Seconds)	21.9	16.6	20.2	14.7	27.7	16.6
Cost per Stop ($)	1.86	1.59	1.77	1.49	2.15	1.59
Cost per Hour ($)	218	154	130	104	300	157
Saving per Hour ($)	64		26		143	
Saving per Year ($)	32,640		13,260		72,930	
Total Saving per Year ($)			118,830			

Table 4. Relative Cost Benefits of Computed Signal Settings

Item	Difference (%)		
	AM	Noon	PM
Stops per 15 min	17.0	5.0	28.0
Cost per Stop	14.0	16.0	26.0
Cost per Hour	29.0	20.0	48.0

A study of Tables 2 and 3 indicates a relationship between cost and traffic volume. As volume, indicated by stops per 15 minutes, increases, so does the saving per hour. The relationship between volume and improvement is supported by Table 1, where relative improvement in each of the measured statistics increases with an increase in volume, or throughput.

CONCLUSIONS

Simulation results of current versus computed signal control for each of the peak periods demonstrates that use of computer-generated signal settings can significantly improve the performance of Garland's traffic network. This performance improvement is measured in terms of throughput, number of stops, wait times, trip times, and queue statistics. Table 5 shows the improvements for the NOON, AM, and PM peak periods which have successively heavier traffic volumes. The larger volumes of traffic in the PM period realize greater relative improvement.

Improvements in network performance also can be translated into estimated costs savings to the motoring public. Using cost figures for stops and delays and potential improvements obtained from the simulation study, yearly cost savings for the three peak periods (i.e., six hours for five days or 25 percent of weekday operation) were estimated as follows:

Table 5. Improvement Factor for Current vs Computed Signal Settings

Item	Improvement (%)		
	Noon	AM	PM
Throughput (Cars)	3.0	4.3	11.5
Queue wait	2.0	13.6	23.0
Average queue wait	33.0	39.0	64.0
Average queue length	11.0	17.0	22.0
Average trip time	18.0	20.0	27.0
Stops	5.0	17.0	28.0

Period	Saving per Year ($)
AM	36,640
Noon	13,260
PM	72,930
Total	118,830

This estimate does not take into account weekends, which constitute 28 percent of the year, and special events which could present very high traffic volumes. Even more importantly, this estimate does not consider the growing nature of Garland's traffic volume.

In addition to the direct dollar benefits, other community benefits would be accrued in the environmental areas of air and noise pollution and in the enhanced safety, convenience, and comfort of daily travel.

Appendix A. NETWORK PERFORMANCE COMPARISON

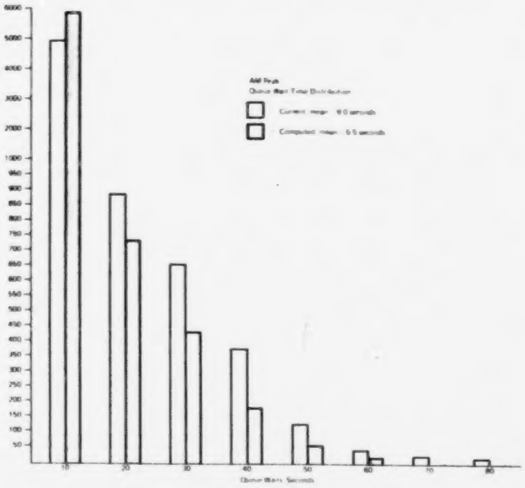


Figure A-1. AM Peak, Queue Wait Time Distribution

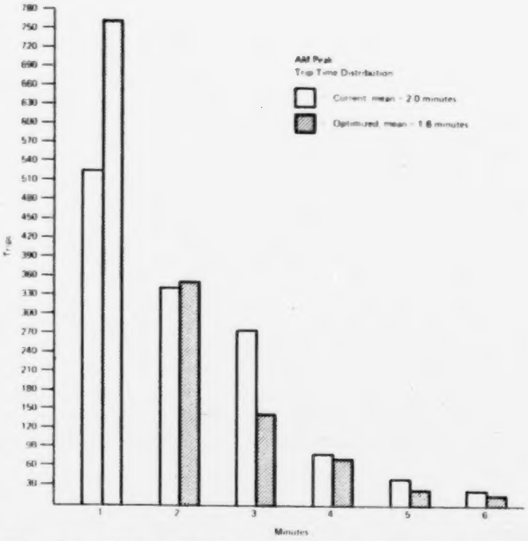


Figure A-3. AM Peak, Trip Time Distribution

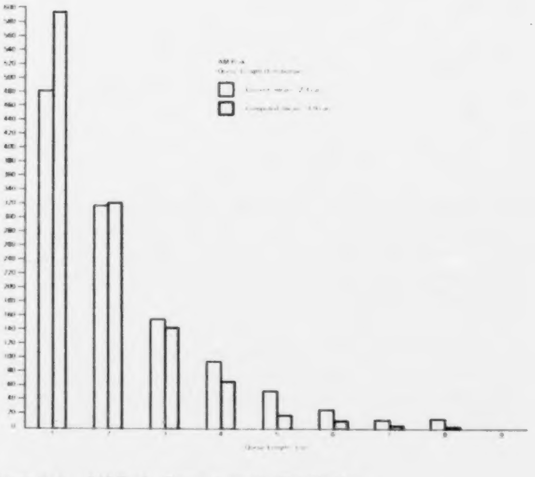


Figure A-2. AM Peak, Queue Length Distribution

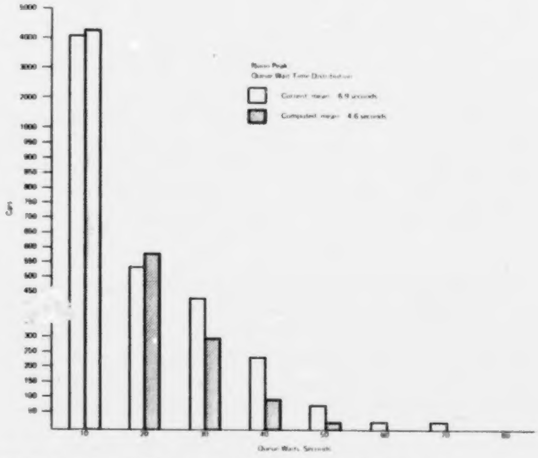


Figure A-4. Noon Peak, Queue Wait Time Distribution

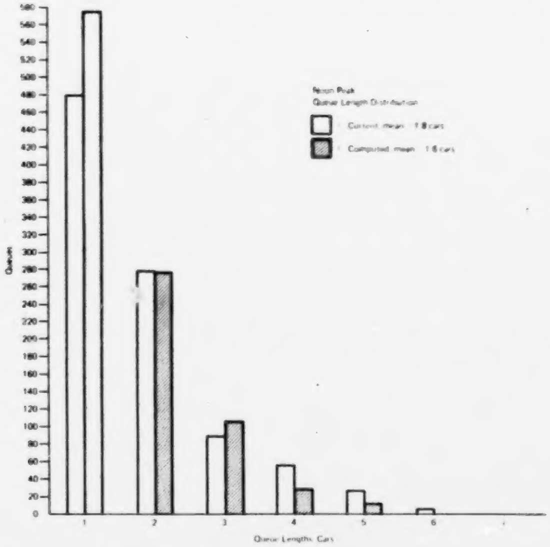


Figure A-5. Noon Peak, Queue Length Distribution

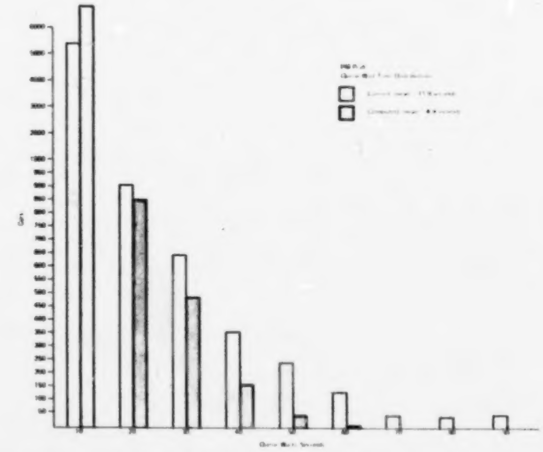


Figure A-7. PM Peak, Queue Wait Time Distribution

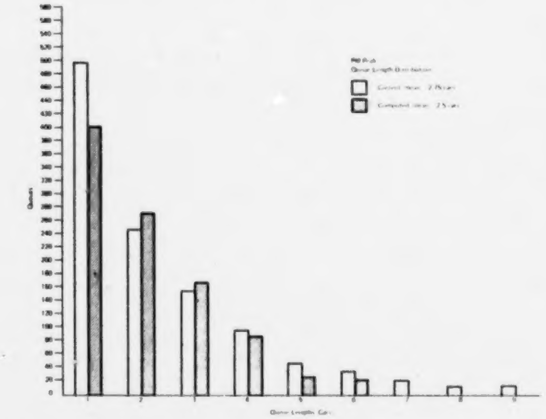


Figure A-8. PM Peak, Queue Length Distribution

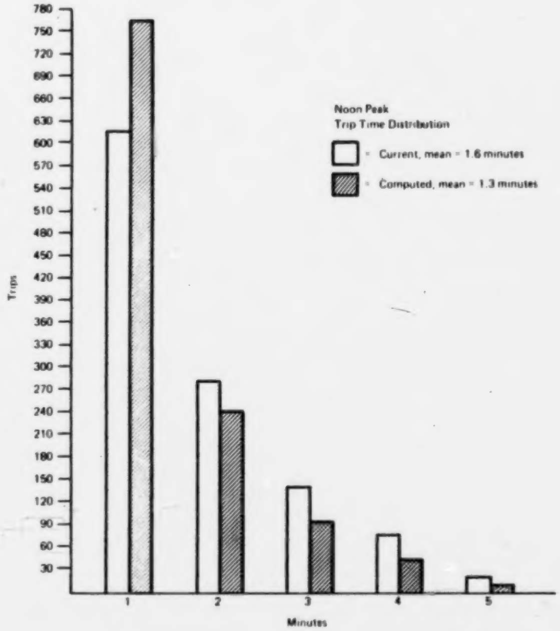


Figure A-6. Noon Peak, Trip Time Distribution

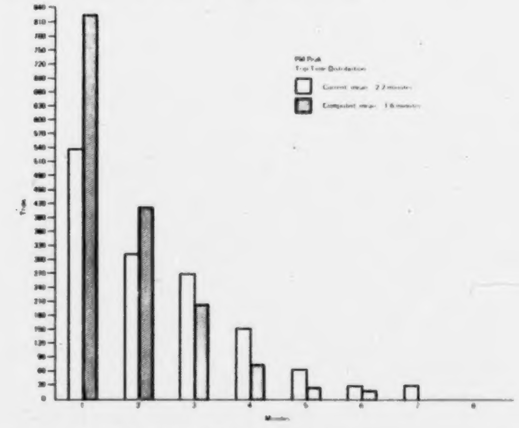


Figure A-9. PM Peak, Trip Time Distribution

Session 11: APL Applications
Chairman: C. Bartlett McGuire, University of California

The use of APL in business and economics has increased significantly in recent years. This session considers APL's simulation features in three financial planning applications and in addition, a number of APL PLUS simulation and optimization algorithms in cash flow management.

Papers
"Corporate Planning Model Design - Computerized Scratch Pads"
Harley M. Courtney, University of Texas

"An Application of Simulation Models to Corporate Planning Processes"
Ronald A. Seaberg, Xerox of Canada, Limited

"APL Models for Operational Planning of Shipment Routing, Loading and Scheduling"
Richard D. Cuthbert, Xerox Corporation

"A Two Asset Cash Flow Simulation Model"
Richard D. Grinold and Robert M. Oliver, University of California

Discussants
Theodore J. Mock, University of California
M. Vasarhelyi, University of California

CORPORATE PLANNING MODEL DESIGN: COMPUTERIZED SCRATCH PADS

Harley M. Courtney
The University of Texas at Arlington

Abstract

This paper describes a modular approach to the construction of financial planning models.

This paper will describe financial planning model design permitting such flexibility of use that the models become management's financial planning "scratch pads." Moreover, the structure of a model designed according to this philosophy will be described and a global run of the model will be presented. Significant interrelationships between model design, operating environment and programming language will be considered.

The corporate planning model has, in recent years, largely supplanted its pedestrian predecessor, the hand-generated budget. An immediate benefit was the freedom from computational constraints. But of fundamental significance is the possibility of elevating the term "profit-planning" from the level of a neologism to that of a concept. "Profit-planning" is used in the financial planning literature to describe anything from preparing budget schedules to select-accounting methods. A significant activity which well-designed planning models permit is the heuristic selection of various investment alternatives so that a myriad of corporate objectives such as planned profits can be more nearly achieved than otherwise. Thus a well-designed model will permit a corporate planner to manipulate the plan almost effortlessly, extracting and inserting segments of operations to determine their effect on the total financial picture.

The majority of planning models have been developed and are still used in the batch mode. It is not surprising that their design has been influenced by the environment in which they have been constructed and used. The typical construction involves a single program (main program) which may call subroutines or may contain all parts of the model within the single program with input and output options provided. A physical problem unrelated to model design, but associated with batch-run models is turnaround time and the physical necessity of handling data cards for each run. In many installations, good turnaround time is a matter of several minutes and typical time is in the hours. If a run is made of the model, and revisions in the financial plan appear to be desirable, then not only must cards for the variables being changed be handled, but all data cards must be rehandled and re-run.

A solution to this problem has been found in the use of time-sharing, but previously unrecognized problems surface with the change in operating environment. These can be attributed to the model structure which, although appropriate for batch-processing mode, appears to be unnecessarily restrictive for a time-sharing environment. The most confining aspect of batch design models is the necessity that each run of the model be a total run, and that output must be specified as a part of the initial input to the model.

The history of the development of a modular, flexible model was that of discovery in stages. A private concern gave the author a corporate planning model to be used for instructional purposes which was designed to run in the batch-mode. While the model had been designed as a structured model into which various firms might adapt their accounting data, it was readily usable for university instruction in financial planning. It was used with some success, but with the previously mentioned problems associated with batch-processing consuming considerable student time. Probably corporate managers would be even more intolerant than students of the time consumption and the start-stop aspects of planning. Since other computer usage in the course was via a time-shared terminal, it seemed that conversion of the model to this mode would assuage student problems in running the model. Turnaround time would be reduced to perhaps twenty minutes. But another problem appeared: If one program was used for the entire model, all variables must be entered for each run of the model, and this would be more onerous than handling cards. For this reason, and in order to simplify the programming by segmenting the task, it was decided that three functions (programs) would be written. The first function would receive input data, the second would perform the calculations, and the third would produce the financial plan. Thus the design of the batch process model and the time shared model is contrasted in Exhibit 1. Note that the batch-processor is holistic in structure while the time-shared model is modular. Another change of substance from the batch to the time-shared model is the change of language, the former being written in FORTRAN and the latter in in APL. Although there were other compelling reasons for the change, it was mandated because (a) the original model was in FORTRAN and APL batch processors are not generally available, and (b) because our time-sharing service provides APL, but not FORTRAN.

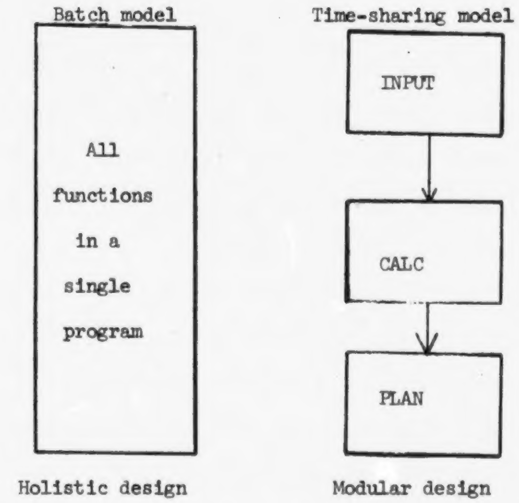


Exhibit 1: Contrasting design characteristics of batch vs. time-sharing models.

BENEFITS DERIVED

Probably the most compelling reason for modular construction of the APL model version was the simplicity in writing functions and debugging them. However, it was discovered that after an initial run of the functions INPUT, CALC, and PLAN, one could then change several input variables most easily by simply redefining them rather than by using the INPUT function again. Then the CALC function could be run and the entire revised plan could be reproduced without requiring use of the lengthy INPUT function. Moreover, if in the initial run of the plan, some one or two output variables appeared to be critical and the additional run was to determine the effect on the critical output variables of changing certain input variables, the entire plan need not be reproduced. Rather, after running the CALC function, given output variables could be obtained by typing the names of the variables. Thus operation of the model was further developed as indicated in Exhibit 2.

At this point design characteristics permitted the model to be used as a "scratch pad" upon which a planning manager could enter incremental changes and view the effects on selected variables in seconds. Typical use of the model in solving financial planning cases was to make an initial run through the model, to examine the plan produced and consider input variables which might be changed to reflect additional investment proposals or asset redeployments. These variable changes were entered individually, CALC was typed and then selected output variables were inspected. Once a number of changes had been made or the plan appeared to be acceptable in respect to the few variables inspected, the total plan would be produced by typing PLAN.

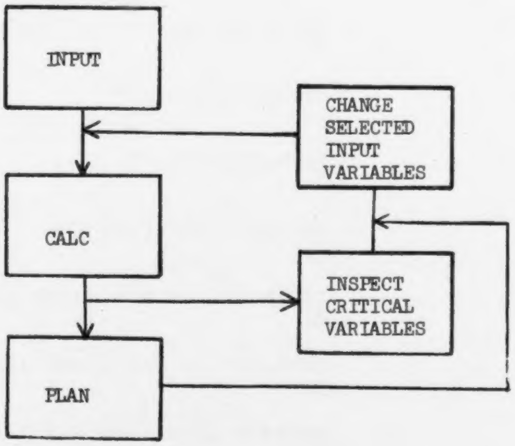


Exhibit 2: Diagram of model operation with modular construction.

The sequence of steps is illustrated on the immediate following pages. The INPUT, CALC, and PLAN functions (programs) are executed in that order; a line has been drawn between each function run for the convenience of the reader. Typing by the user is indented six spaces with the exception of literal input which begins at the left margin. Computer-controlled typing begins at the left margin except as otherwise programmed. For example the first function is called by typing the name INPUT which is indented. The response is a program inquiry, "NAME OF CO?" Since literal input is required, the terminal requests input at the left margin. In contrast, the next inquiry regarding the number of periods and year is numeric and the terminal indents six spaces to receive the response.

Exhibit 3 is the first run of a three year financial plan. All significant corporate objectives would be met if the plan proposed were realized. However, some additional funds must be obtained (see last current liability item), and

```
      INPUT
NAME OF CO?
BARCO ENTERPRISES, INC.
NO.PERIODS OF OUTPUT AND FIRST YEAR OF OUTPUT?
⎕:
      3 1973
RATIO OF CASH TO SALES?
⎕:
      .04
MARKETABLE SECURITIES?
⎕:
      5000
YIELD ON MARKETABLE SECURITIES?
⎕:
      .05
RATIO OF ACC REC TO SALES?
⎕:
      .09
BEGIN RAW MAT OF 1ST PERIOD?
⎕:
      0
RAW MAT TURNOVER?
⎕:
      4
TURNOVER OF WIP?
⎕:
      18
DEGREE COMPLETION WIP?
⎕:
      .5
BEGIN INV OF FIN GOODS (1ST PERIOD ONLY)?
⎕:
      0
TURNOVER OF FIN GOODS INV?
⎕:
      10 12 12
PREPAID EXP/SALES(N+1)?
⎕:
      .02
```

Exhibit 3: Illustration of a run of the principal functions.

```
OTHER ASSETS?
⎕:
      5400
PLANT AND EQUIP?
⎕:
      300000 300000 500000
BEGIN BAL OF ACCUM DEPR?
⎕:
      0
DEPR RATE?
⎕:
      .075
LAND?
⎕:
      20000
DEFERRED CHGS?
⎕:
      2400
ACCTS PAY/MAT. PUR. ?
⎕:
      .12
DIVIDENDS PAY/DIV DECLARED?
⎕:
      .08
DIV PAYOUT RATE
⎕:
      0 0 .3
WAGES PAY/DIRECT LABOR COST?
⎕:
      .05
TAXES PAYABLE?
⎕:
      1000 1000 1500
SHORT TERM LOANS?
⎕:
      35000 35000 10000
INT RATE ON ST NOTES?
⎕:
      .09
LONGTERM DEBT MATURING WITHIN 1 YEAR?
⎕:
      0 0 5000
DEFERRED TAXES?
⎕:
      0 0 4000
LONG TERM DEBT?
⎕:
      0 0 45000
INT RATE ON LT NOTES?
⎕:
      .06
COMMON STOCK SHARES ISSUED?
⎕:
      2500
PAR VALUE COMMON SHARE?
⎕:
      10
PREMIUM ON STOCK BALANCE(S)?
⎕:
      20000
BEGIN RETAINED EARNINGS?
⎕:
      0
YOU MAY ENTER AFIRST PERIOD SALES AMT. AND A GROWTH RATE OR
YOU MAY ENTER ABSOLUTE VALUES.
```

```
DO YOU WISH TO USE A GROWTH RATE?
NO
ENTER SALES VALUES FOR PERIODS OF OUTPUT DESIRED + ONE
□:
      50000 100000 225000 280000
COST OF SALES/SALES? ENTER EITHER A SINGLE VALUE OR
VALUES FOR NO. YEARS OUTPUT DESIRED + ONE.
□:
      .6 .55 .5 .5
DIRECT LABOR COST/COST OF GOODS COMPLETED?
□:
      .4
MATERIAL COST/COST OF GOODS COMPLETED?
□:
      .25
OVERHEAD/COST GOODS COMPLETED?
□:
      .35
FIXED INDIRECT LABOR?
□:
      8000 8000 17000
OTHER OUT OF POCKET FIXED OVERHEAD?
□:
      5000
VARIABLE OVERHEAD/DIRECT LABOR COST?
□:
      .3
VARIABLEADMINISTRATIVE EXP/   SALES?
□:
      .04
FIXED ADMIN EXP?
□:
      5000 10000 20000
VARIABLE SALES EXP/SALES?
□:
      .06 .06 .05
FIXED SELLING EXPENSE?
□:
      5000 10000 25000
EXTRAORDINARY GAINS?
□:
      0 0 2000
EXTRAORDINARY LOSSES?
□:
      0 4300 0
END OF INPUT REQUIREMENTS
TO PROCEED, TYPE 'CALC'
```

```
      CALC
CALCULATIONS ARE BEGINNING, PLEASE STAND BY.
CALCULATIONS COMPLETED.
TO PRODUCE THE COMPLETED PLAN,TYPE PLAN
```

```
      PLAN
TO CENTER OUTPUT VERTICALLYON THE PAGE, ROLL THE PAPER
FORWARDTOA NEW PAGE AND THEN PRESSTHE RETURN KEY.  IF
CENTERING IS NOT DESIRED, SIMPLY PRESS THE RETURN KEY.
```

Exhibit 3: Continued

BARCO ENTERPRISES, INC.
BALANCE SHEET
FOR YEAR ENDS

	1973	1974	1975
CURRENT ASSETS			
CASH	4,000	9,000	11,200
MARKETABLE SECURITIES	5,000	5,000	8,126
ACCOUNTS RECEIVABLE	4,500	9,000	20,250
RAW MATERIALS	3,438	7,031	8,750
WORK IN PROCESS	986	1,635	3,189
FINISHED GOODS	5,500	9,375	11,666
PREPAID EXPENSES	2,000	4,500	5,600
TOTAL CURRENT	25,424	45,542	68,781
FIXED ASSETS			
PLANT AND EQUIPMENT	30,000	30,000	50,000
LESS ACCDEPR.	2,250	4,500	8,250
LAND	20,000	20,000	20,000
TOTAL FIXED	47,750	45,500	61,750
OTHER ASSETS	5,400	5,400	5,400
DEFERRED CHARGES	2,400	2,400	2,400
TOTAL ASSETS	80,974	98,842	138,331
CURRENT LIABILITIES			
ACCOUNTSPAYABLE	1,478	2,198	3,650
DIVIDENDS PAYABLE			733
WAGES PAYABLE	710	1,178	2,296
ACCRUED TAXES	1,000	1,000	1,500
SHORT TERM LOANS	35,000	35,000	10,000
LT DEBT MATURING			5,000
FUNDS NEEDED	2,771	14,700	
TOTAL CURRENT	40,959	54,075	23,179
LONG TERM LIABILITIES			
LONG TERM DEBT			45,000
DEFERRED TAXES			4,000
TOTAL LIABILITIES	40,959	54,075	72,179
STOCKHOLDERS EQUITY			
COMMON STOCK	25,000	25,000	25,000
PREMIUM ON STOCK	20,000	20,000	20,000
RETAINED EARNINGS	-4,985	-234	21,152
TOTAL DEBT +EQUITY	80,974	98,842	138,331

Exhibit 3: Continued

there is some concern that the company may have difficulty placing the long-term debt to be issued during 1975, given the low current ratios for 1973 and 1974. Consequently, a proposal to lease $20,000 of the equipment during 1973 and 1974 rather than purchasing it is being considered. The leasing cost will be $6,000 the first year and $5,000 the second. New equipment will then be purchased in the third year. By leasing equipment, short term loans required in 1973 can be reduced and the necessary additional funds for 1974 are less. Since additional funds must be

```
                     BARCO ENTERPRISES, INC.
                       INCOME STATEMENT
                     FOR THEFISCAL YEARS

                              1973          1974          1975
                           ----------    ----------    ----------

SALES                          50,000       100,000       225,000
COST OF GOODS SOLD             30,000        55,000       112,500
                           ----------    ----------    ----------

STD GROSS PROFIT               20,000        45,000       112,500
OVERHEAD VARI.                  7,085         1,709          ¯652
                           ----------    ----------    ----------

ADJ. GROSS PROFIT              12,915        43,291       113,152
OPERATING EXPENSES
VARIABLE ADMIN. EXP             2,000         4,000         9,000
FIXED ADMIN. EXP                5,000        10,000        20,000
VARIABLE SELLING EXP            3,000         6,000        11,250
FIXED SELLING EXP               5,000        10,000        25,000
                           ----------    ----------    ----------

OPERATING INCOME               ¯2,085        13,291        47,902
INTEREST EXPENSE                3,150         3,150         3,900
INVESTMENT INCOME                 250           250           250
EXTRAORDINARY ITEMS
EXTRAORDINARY GAINS                                         2,000
EXTRAORDINARY LOSSES                          4,300
                           ----------    ----------    ----------

INCOME BEFORE TAX              ¯4,985         6,091        46,252
FED. INCOME TAX                               1,340        15,701
                           ----------    ----------    ----------


NET INCOME                     ¯4,985         4,751        30,551
                           ==========    ==========    ==========
```

```
                     MANUFACTURING OVERHEAD
                      FOR THE FISCAL YEARS
                              1973          1974          1975
                           ----------    ----------    ----------

FIXED INDIRECT LABOR            8,000         8,000        17,000
OTHER FIXED OVERHEAD            5,000         5,000         5,000
DEPRECIATION                    2,250         2,250         3,750
VARIABLE OVERHEAD               4,260         7,065        13,775
                           ----------    ----------    ----------

  TOTAL ACTUAL OH              19,510        22,315        39,525
  OVERHEAD APPLIED             12,425        20,606        40,177
                           ----------    ----------    ----------

UNDER ¯OVER APPLIED             7,085         1,709          ¯652
                           ==========    ==========    ==========
```

Exhibit 3: Concluded

secured for the second year, management is also considering the issuance of more shares in 1973. The planner then "scratch-computed" by first considering the effect of leasing equipment and reducing short-term debt, and then calculating the additional shares required to cover the funds deficit. The steps as illustrated in Exhibit 4 are:

1. Redefine the values for plant and equipment, other fixed overhead, and short term loans.
2. Compute new plan values using CALC.

```
      PLTEQ←10000 10000 50000
      OFIXOH←11000 10000 5000
      STLOANS←15000 15000 10000

      CALC
CALCULATIONS ARE BEGINNING, PLEASE STAND BY.
CALCULATIONS COMPLETED.
TO PRODUCE THE COMPLETED PLAN, TYPE PLAN

      TCA÷TCL
1.010533173  1.108029534  2.424549529
      FUNDNEED
6971.111111  21726.49167  3900.331407
      NI
¯7685  3425.175  30551.08333

      CSNO←CSNO+500
      PRECS←PRECS+8×500

      CALC
CALCULATIONS ARE BEGINNING,PLEASESTAND BY.
CALCULATIONS COMPLETED.
TO PRODUCE THE COMPLETED PLAN, TYPE PLAN

      FUNDNEED
0  12726.49167  0
      TCA÷TCL
1.509415808  1.418677585  3.052539434

      STLOANS[2]←STLOANS[2]+13000

      CALC
CALCULATIONS ARE BEGINNING, PLEASE STAND BY.
CALCULATIONS COMPLETED.
TO PRODUCE THE COMPLETED PLAN, TYPE PLAN

      FUNDNEED
0  639.0916667  0
      TCA÷TCL
1.509415808  1.379461447  3.01316769
      MKTSC
7028.888889  5000  9187.068593

      STLOANS[3]←STLOANS[3]-4000
```

Exhibit 4: Illustration of the "scratch pad" characteristics of the time-sharing model.

3. Inspect some critical variables - current ratio (total current assets divided by total current liabilities), net income, and funds needed.
4. Redefine values for common stock and premium.
5. Compute new plan values using CALC.
6. Inspect the critical variables.
7. Continue until acceptable variables appear.

When the user is satisfied that the critical variables are satisfactory, he will then proceed to production of the entire plan for perusal. Note that the turnaround time is governed by the rapidity of the user's mental processes rather than by model constraints since run time for CALC is usually equal to the time required for the typewriter to type the messages involved. The

```
      STORE
PLAN OUTPUT IS STORED

      ARSALE←.09 .15 .15
      SALES←50000 120000 265000 330000
      VADEXPSAL←.04 .05 .05

      CALC
CALCULATIONS ARE BEGINNING, PLEASE STAND BY.
CALCULATIONS COMPLETED.
TO PRODUCE THE COMPLETED PLAN, TYPE PLAN

      PLANDIF
TO CENTER OUTPUT VERTICALLY ON THE PAGE,ROLL THE PAPER
FORWARD TO A NEW PAGE AND THEN PRESS THE RETURN KEY.  IF
CENTERING IS NOT DESIRED, SIMPLY PRESS THE RETURN KEY.

                    BARCO ENTERPRISES, INC.
                         BALANCE SHEET
                         FOR YEAR ENDS

                            1973          1974          1975
                         ----------    ----------    ----------
CURRENT ASSETS
CASH                            800         1,600         2,000
MARKETABLE SECURITIES                                    ¯3,126
ACCOUNTS RECEIVABLE                         9,000        19,500
RAW MATERIALS                   688         1,250         1,563
WORK IN PROCESS                  31           321           567
FINISHED GOODS                1,100         1,667         2,083
PREPAID EXPENSES                400           800         1,000
                         ----------    ----------    ----------
   TOTAL CURRENT              3,018        14,638        23,587
                         ----------    ----------    ----------
FIXED ASSETS
PLANT AND EQUIPMENT
LESS ACC DEPR.
LAND
                         ----------    ----------    ----------
   TOTAL FIXED
OTHER ASSETS
DEFERRED CHARGES
                         ----------    ----------    ----------
   TOTAL ASSETS               3,018        14,638        23,587
                         ==========    ==========    ==========
```

Exhibit 5: Illustration of STORE and PLANDIF functions permitting the difference between two plans to be produced.

messages were added since response time may be up to ten seconds if many users are on the system, and students, being accustomed to faster responses, were reassured by them.

Some additional modules have been added to the model and other additions are anticipated. Two functions have been added which permit the user to store the results of the current run, to make changes in variables and make a second run. Then the second run can be produced and/or the differences between the first and second runs can be produced (the plan production will have positive and negative values representing changes between the successive plans). This feature is illustrated by Exhibit 5. Another set of modules permit the user to specify certain financial goals

```
        GOALS
ENTER THE UPPER LIMIT FOR THE CURRENT RATIO
⎕:
        2.3
ENTER THE LOWER LIMIT FOR THE CURRENT RATIO
⎕:
        1.2
ENTER THE UPPER LIMIT FOR THE DEBT/EQUITY RATIO
⎕:
        .5
ENTER THE LOWER LIMIT FOR THE DEBT/EQUITY RATIO
⎕:
        .2 .2 .35
ENTER THE TARGET PROFIT ON SALES DOLLAR
⎕:
        ¯.1 0 .1
TARGET ASSET TURNOVER
⎕:
        .8 1.2 1.5
TARGET EARNINGS PER SHARE
⎕:
        ¯2 1 5
END OF INPUT FOR GOALS

        GOALVAR

CURRENT RATIO GREATER THAN UPPER LIMIT IN 1975 BY 1.1

PROFIT ON SALES DOLLAR DEVIATED FROM THE TARGET BY
                      ¯0.054  0.025  0.037
FOR THE 3 YEARS RESPECTIVELY

ASSET TURNOVER DEVIATED FROM THE TARGET BY
                      ¯0.025  0.022  0.128
FOR THE 3 YEARS RESPECTIVELY

EARNINGS PER SHARE FELL SHORT OF THE TARGET BY
      0.56  0.16  FOR 1973 1974
EARNINGS PER SHARE EXCEEDED THE TARGET BY
      5.26 FOR 1975

ALL OTHER GOALS WERE ACHIEVED.
```

Exhibit 6: Use of programmed goals to diagnose acceptability of the plan.

such as earnings per share and current ratio and to determine the extent to which these goals have been met. Thus critical variables may be specified in the model and their achievement evaluated with each successive pass through CALC. For some purposes this is a more efficient "scratch pad" approach than the informal inspection of individual output variables. The operation of this set of functions is illustrated in Exhibit 6.

The expanded planning model and the alternative use choices give the user maximum flexibility in that he may:

1. Proceed straight through the basic three functions, INPUT, CALC, and PLAN.
2. Input the basic data, perform calculations, inspect critical variables, and either recycle or produce the plan.
3. Input data, perform calculations, input goals, and then examine goal achievement, followed by a recycle or plan

production.

4. Input data, perform calculations, store the plan, recycle by changing input variables, performing calculations, and then determining the differences between the first and second run of the plan.

There are other variations of course, with this variety essentially giving management a powerful scratch pad for planning financial operations.

Several other features can be conveniently added due to the modular construction. RESTORE, a function which restores a stored plan so that it can be produced can be easily added. In fact, a series of store and restore functions can be created and distinguished by adding 1,2, or 3 to the function name. The successive runs of the plan could be retained within the same workspace for future use or reference. (In many instances it will be more convenient to save the results of a run in a separate workspace.) A function called QUARTERS would convert annual values (such as interest rates and programmed fixed costs) to quarterly values, thereby permitting the generation of plans by quarters in addition to years. Some additional segmentation of the existing functions may be necessary to accomplish this, but the concept should be transparent.

THE RELATIVE MERITS OF APL

Although other languages are available via time-sharing, APL possesses several characteristics which recommend it for use. Some of these are the extension of arithmetic operations to vectors on an element by element basis and the ability to mix integers and vectors. This ability eliminates the necessity for loops when combined with the use of certain primitive functions such as take (↑) and drop (↓) which are useful in accomodating lead and lag relationships. Moreover, subscripting of variables is unnecessary. In the model, for example, cash at the end of a fiscal period is assumed to be a function of the following year's sales. The program statements in FORTRAN and APL are compared below:

```
    FORTRAN:
          DO 205, I=1,NOYRS
          CASH (I) = RCASSAL(I)*SALES(I+1)
    APL: CSH←RCSHSALE×¯PERIOD↑SALES
```

Of course the 'DO' loop encompasses variables other than cash. But it should be noted that while several loops are required in FORTRAN, none are required in the APL version of the model due to the structure of the language.

Another significant advantage of APL is that no dimension statements are required; in contrast, the FORTRAN version of the model required fourteen lines of such statements. Output formatting is also simpler.

```
    FORTRAN:
          WRITE(108,2040) (CASH(I), I=1,NOYRS)
          2040 FORMAT ($CASH$,16X,F8.0,4(3X,F8.0))
    APL: 'CASH         ','BCI14' ΔFMT CSH
```

The lack of dimensioning in APL means that the model can operate on one, five, twenty, fifty year projections, while the FORTRAN model is constrained to a five year projection. The formatted plan output in either case is limited to the width of the printing device.

CONCLUSIONS

The conversion from hand-generated financial budgets to computerized planning models was a significant step in the development of corporate management. The concept of multiple cuts of the budget became a reality rather than a step resorted to on a partial basis or only in the most extreme situations. The holistic design of such early planning models was appropriate for the batch operating environment.

But with the availablilty and use of timesharing, design characteristics should recognize and capitalize on changes in the operating environment. While some have called attention to desirable characteristics of models, it is also important to recognize interrelationships between the operating environment, model design characteristics, and even the choice of programming language. For example, some have advocated the use of flexible rather than structured models (as the one described here). But flexible models begin to assume the characteristics of a limited programming language, and require more user time than structured models. Thus if one utilizes a concise language (APL) for model construction, the cost of adapting a structured model to unique circumstances may be less than the cost of using (overlooking the greater programming cost) a less powerful language.

A modular-designed corporate planning model such as described here provides management with a scratch pad approach to planning, permitting attention to proposals and their effects rather than distraction by interruptions occasioned by total runs of plans when such is unnecessary. The ability to enter only those variables being changed and to inspect the effects of the changes within five to ten seconds permits the type of concentration being sought for by researchers in computer assisted instruction. When this state is achieved in corporate planning, the computer has become an interactive management tool rather simply a rapid calculator. The difference is qualitative as well as quantitative, since management can do in an appropriately designed interactive environment what cannot be accomplished in a batch oriented environment due to time constraints and limited retention ability of the human mind.

"AN APPLICATION OF SIMULATION MODELS TO CORPORATE PLANNING PROCESSES"

Ronald A. Seaberg
XEROX of Canada Limited
January 1973

A family of timeshared computer models written in APL have been developed in an effort to link the functional areas for communication, planning and control purposes. The models incorporate the concepts and tools of simulation, forecasting, long-range planning and probabilistic budgeting. Developed in a short time span and at low cost, they are widely used at XEROX in both corporate and region offices in both the U.S. and Canada. Under development is a system of statistical and econometric models. The paper discusses the approaches used to design, implement and involve the functional managers in building and utilizing the models.

INTRODUCTION

We have developed and are utilizing a series of deterministic simulation models to assist us in financial planning for Xerox (XCL) and additionally as a step in the development of corporate planning models.

The purpose of this paper is to describe how we use the models in our planning process, the types of models we utilize, the initial implementation procedure and how we maintain the models in a changing internal/external environment.

These models are on a timeshared (T/S) computer, were developed by the functional users, and are written in APL. They access an extensive series of on line data bases which mechanically interact with batch system data bases.

Not at issue is whether or not simulation is of value. We accept and have demonstrated to ourselves that it can make a significant contribution. Rather, we are essentially concerned with applications to which simulation models can make an economic contribution and how to implement them in a way that managers (without a management science background) will be able to incorporate them into their planning and decision-making processes.

The definitions of models and simulation are two popularly raised questions when working with corporate managers. Models are representations of systems which themselves are too large to be brought into a laboratory or otherwise experimented on in their natural environment (Ackoff 1970). Models can be classified as physical (ships in tow tanks), graphic, symbolic representations (algebraic equations) of the system; or a specified procedure for the evaluation of some criterion such as expenses or profit (Schweyer) that depend on other operating conditions subject to control by management.

Simulation is the manipulation of a model in such a manner as the "properties" of the system can be studied. In the sense I'm using it, simulation is an experiment or manipulation of a model that reproduces XCL operations (a functional component or the whole firm) as it moves through time. The manipulation may be by hand, computer, or by a combination of man and computer working together. The simulation models we have worked with to date have been essentially of the latter type (man-machine) although we have built some forecasting models that can 'stand alone' under strict computer control. We simulate systems (such as the operations of XCL) because we want to understand how they work, determine the factors that influence their behavior and observe how they react to changes in their environment as an aid to planning, forecasting, and other decision processes.

SIMULATION MODELS BEING USED IN FINANCIAL PLANNING AT XCL

It is most important to realize that we do not have a planning model as such, rather, we have a series of models which are changing as we learn more about our business and as we learn more about how models can assist us in improving our planning and decision-making capabilities.

Simulation models have been previously (Gershefski 1971) classed into three types - budget compilers, simple mathematical models, and large complex integrated models. We use all three types and variations in between.

Our first simulation model was of the simple mathmatical type, it was developed in crude operational form within 1 calendar month with the expenditure of less than $3000.00. This model was built to assist us in the development of our rolling 12 -month forecast of our business, a task which is performed monthly and is used in resource planning, profit planning, as a control device, and in the development of our operating plan. Prior to the model, this activity required about 4 man-weeks of analyst level effort.

This initial model was operated through a terminal where the analyst and manager (generally both) controlled 30 variables (such as growth rates, product trading relationships, productivity, etc.) and obtained a forecast of 300 output variables per month for 12 months (such as order forecasts, inventory changes, resource requirements, revenue, expense and profit contribution, etc.) (Seaberg 1972). This initial system can be seen in Chart 1 (Initial Simulation System). On this chart you can see the interrelationship of the timeshared data base with the in-house batch systems. Initially the data was keyboard entered into the T/S data base; now, there are mechanical (data tape) transfers .

To summarize our initiation to simulation models, one could say that it started with a recognition that a computer model could be of significant assistance in obtaining a more timely and consistent forecast of XEROX operations at a significant reduction in cost (4 man-weeks to 2 man-machine days) and infinitely more flexibility in varying assumptions and observing the simulated outcome on activity, revenue, expense variables.

We classified this initial model as a simple model. While it simulated our marketing function in a fair amount of detail, our distribution, service (support) and financial functions were handled in highly aggregated and simplistic procedures (although in more detail than the manual procedure which the model replaced). The model did however, stress key interrelationships in both our financial and operating structures and for this last reason we later used this particular simulation model for the basis of a model designed to assist in long range studies and projections.

Our long range planning cycle (10 years) begins with a projection of current trends of key variables. Management then may change these key variables consistent with what they feel to be desired directions (objectives and goals) and the models are then executed to determine the effects on operations and the resulting financial implications.

INTEGRATED MODELS

The requirement for increased model complexity becomes apparent as we begin examining causal factors; the interrelationships within, as well as between the functional areas; and for the development of operating plans and budgets.

Beginning with our simple model, we, together with the functional areas, enlarged the models to describe in more detail, their operations and the financial procedures of XCL. From this effort we derived more complex, integrated models for use in forecasting , planning, and simulating procedural changes (operations or financial) and for special one-time or infrequent studies such as pricing reviews, promotional campaigns, and facility locations analysis. As can be expected these large models have both many more input variables (compensation levels, trends by employee group, freight rates by destination and point of origin, etc.) and outputs (an entire operations and financial plan). These larger models tend to be operated primarily by analysts supporting the functional manager.

It is difficult if not impossible in both planning and forecasting to avoid looking outside the firm to the external environment. Typically and in the case of XCL many of these relationships are not clearly defined and even more rarely quantifiable. However, these forces cannot be ignored and must (even if only implicit) be incorporated in a forecast and plan. Typically we find that these exogenous factors for planning purposes, are assumed not to change or at most, that changes in them will have only a minimal impact on the firm. However, on a hind-sight basis it is nearly always demonstratable that environmental forces both economic (up or downswing) and social (riots) did have an influence on the firm (generally earnings). It becomes increasingly important to provide an explicit as possible a mechanism to incorporate these influences in our planning process.

To this challenge we are developing other models and information links to be incorporated into our planning system. Partial to-date results from these efforts include a series of statistical and econometric models and data bases.

ECONOMETRIC MODELS

To date, we have focused primarily on the development of "search-analysis-display" timeshared programs to examine econometric series (GNP, unemployment, interest rates) and XCL DATA (sales, profits) and applying the statistical concepts of regression analysis (single, multiple and stepwise), find and determine relevant relationships. The Service Bureau (which supports APL) provides an economic data base of 6000 series maintained by Statistics Canada (CANSIM) which in turn is the central agency in Canada for funnelling economic data from originating government agencies (agriculture, commerce, etc .).

A major problem we encountered in doing this type of project is organization of the data such that both corporate and economic series can be retrieved quickly and stored on a comparable basis for use in the analytical models. This is a major problem since these are on-line models and the analyst wants to spend only minimal time retrieving and structuring data so that it is in comparable form. We overcame this obstacle and an example is demonstrated on Chart 2 (A Procedure for Structuring Data Series With Timesharing).

Through the terminal the user specifies both the economic series (CANSIM code) and the XCL data which the computer retrieves from the files.

Another series of programs shapes the data according to information entered through the terminal by the analyst covering such things as time periods to be utilized, lead and lag relationships, and the identification of dependent and independent variables. Other programs then take over and check for data completeness, deletion of any extraneous data, and that the data is structured properly for the analysis programs.

We have obtained national economic forecasts from the Institute of Policy Analysis, University of Toronto and these forecasts also reside on-line for use with the planning and simulation models. A sample output of an economic analysis and projection related to XCL data is contained on Chart 3. While this analysis results from a multiple regression analysis, we are developing more sophisticated models (to be covered in Future Projects section). These print programs can also be used by market research to show market or geographical profiles at a point in time or how they are changing over time in addition to comparison with economic variables.

STATISTICAL MODELS

As we were completing our initial simulation models, it was becoming apparent that a variety of statistical forecast models would significantly enhance the value of the simulation models especially if they were equipped with an override capability to allow the manager to easily add information affecting the outcome not contemplated in statistical analysis. These statistical models were designed to be used for estimating input variables with both the planning and forecasting (12 month rolling outlook) simulation models as well as on a stand alone basis.

Late in 1971, exponential smoothing models were made operational. Three models were included in the general model - seasonal, single and double. Optimization procedures are an integral part of the models and are used to compute demand levels, trends, seasonals, and the alpha weighting factors. A tracking system for both previous actuals and forecasts is included to continually (over time) determine the optimal forecasting model (criteria is the minimum absolute deviation - MAD). In addition to the MAD, a forecastability index is also produced (actual performance ÷ standard deviation). Chart 4 shows a typical output of these models.

The effect of introducing this set

of statistical models was something only short of fantastic. The forecast MAD was decreased by an average of 30% and in some instances by 50%. The other significant effect was a significant reduction in the time to prepare a reliable forecast since now the simulation models could be operated with minimal human intervention. It is significant to point out that if managers massage the statistical forecasts to incorporate special knowledge, such as sales campaigns, inventory constraint, (a man-statistical-simulation interface) the MAD decreases by a further 10 per cent indicating the importance of managerial involvement and his insight into the process. I would point out that the computer based simulations models without the statistical models predicted about as well as the purely manual models (the great benefit of course being time and human effort savings). This is, of course, because the initial simulation models were designed directly from the manual process they were to replace.

For use with the statistical models a series of curve fit programs are available. We have found these in many instances to be excellent (although naive) forecast models for some econometric series. Chart 5 shows one way in which we use them in our planning process.

We are currently developing other statistical forecasting techniques which will be covered in the future projects section.

BUDGET COMPILERS

Once we have a set of simulation results from the XCL operations models we execute a set of budget compiler models that allocate resources to organizational units and programs. The output includes budgets and pro-forma financial statements. Most of the budget model input variables are directly taken from the results of the operating and financial simulation models. A few input variables are controlled separately such as depreciation write-off assumptions and allocation rates for overhead expenses. As in the other simulation models a manual override capability exists for any last minute "management directions" .

THE ROLE OF MODELS IN THE CONTROL FUNCTION

An important role of the above models and their data bases has been the derivation of a series of "control" models. The establishment of a control system is one important component of the planning process that provides a feedback mechanism to assure that objectives/goals are being achieved. Chart 6 shows how the feedback system works.

As actual performance data is obtained, it is incorporated in the forecasting simulation models (through the statistical analysis models) which in turn are a feedback loop to the planning models.

In the control function, a discrepancy between actual observations and previously defined objectives (Plans)is generally the stimulus to trigger management action. The statistical forecasting models (discussed above) utilizing actual performance data, projects trends and in addition do a comparison against plan. These "actual-forecast-plan" relationships are made available either through routine or exception reports. While these control models are passive in the sense that they do not trigger action by themselves to correct perceived deviations, they do provide the manager with the exception information which, when coupled with the simulation models, provides the manager with an adaptive mechanism for positive action. Over the past two years each of our functional areas have made extensive use of the simulation models in this control context.

UPDATE AND REVISIONS OF PLANS

Planning is a continuing process. At various time intervals a plan must be adopted as the basis for objectives and resource allocation at which time it becomes an Operating Plan. However, every plan is based on assumptions, and if the assumptions contain gross defects, then we have an invalid plan which can be worse than no plan at all (especially from the viewpoint of credibility). It may then be necessary within the operating cycle to revise and update the Plan. The models are used in this process which in turn stimulates changes to the long range plans which are then simulated and adjusted for this change in current direction.

THE PLANNING SYSTEM WE ARE USING

This next chart (chart 7) puts together all of the various models we have been discussing above (APL in Business--System at 28 months). As is suggested by the chart we have to a very large degree modularized the models so that they could be used interchangeably or on a stand alone basis. The data base contains operating plans, long range plans, forecasts, current and historic operating results. About 99% of the data is either derived from the models or in the case of actual operating data, is mechanically fed from batch systems. The models to the left of the Data Base are the models discussed in this paper.

The special studies are models that have been developed by the functional areas for use in additional analysis of the data and "one-time-only" projects. 95 percent of these programs are written (in APL) directly by the functional department.

The Reporting System indicated in the upper right hand side is a series of programs to format the data or information in a variety of user defined ways. A unique function within the APL system allows the user to develop some basic reports in just a few seconds. Elaborate reports take longer. This is particularly useful with the simulation models as frequently a unique problem is being examined and the manager or analyst does not want to wade through predetermined general reports to observe the effects of a particular set of simulation outputs. In addition this flexible report writing procedure makes it very easy to change report formats .

One of the chief advantages of this type of an overall system is that it links together the historical and future data, and all of the parts of the business so that the business can at the same time be viewed as an integrated whole or be broken down into its various components. Through the simulation models , it can be readily ascertained how a decision in one functional area will reverberate through the system and affect other functional areas and the firm as a whole. These effects can be simulated and shown in seconds or minutes rather than hours or days for a whole series of alternatives. Through the use of "base cases", variable sensitivity analysis is quickly calculated.

FUTURE PROJECTS

In the same sense that planning is an on-going process so are these models and data bases. We find that solutions to one problem generally result in additional questions or demonstrate a lack of understanding of other processes. For us this means that our models must change to answer new questions and to handle problems and alternatives not previously considered. Little, probably best described this process as an "analysis-education-decision process" where man in working with models updates his intuition as he understands more about the problem and the models assist in this process by interrelating the factors for him (Little 1970).

We process model changes on an on-going basis by changing, discarding and recreating, or at the same time both enlarging some models (to account for added complexities) and adding simplifying models for summary level analysis.

We are also continuously researching new capabilities in order to learn more about our business, its environment, and to assist in developing better ways of doing business.

On Chart 7 the asteriks represent some of the projects with which we are currently involved.

ECONOMIC DATA ANALYSIS

As previously indicated, we already have both historical and forecasted data for several economic series and a broad set of statistical routines for analysis. Currently under development are I/0 models to develop industry and geographic forecasts, models to combine both economic and market research data (potentials, penetration growth rates). From this will come (hopefully) information for better resource allocation, identification of growth areas and a measure of the influence of environmental data on our business.

PROBABILISTIC FORECASTING

We are already doing statistical variance analysis as a part of our statistical and feedback models discussed earlier and this provides us with the capability of applying confidence intervals to our statistical forecasts in addition to the point estimates.

We are currently working on the capability of doing risk analysis utilizing "Monte Carlo" simulation methods (AMA 1972). The objective here is to integrate modeling, probability theory and simulation for investment analysis.

SENSITIVITY ANALYSIS

Currently we have the capability of measuring the sensititivity of inputs by changing an input variable, re-simulating and mechanically measuring the change in output variables against a previously defined "base case". This method while it is operational - it is crude. We are enhancing the existing capability with likelihood information for the input variables.

THE IMPLEMENTATION PROCEDURE

We became involved with simulation and models when we were able to demonstrate to ourselves and management that this alternative provided us with better forecast and planning procedures than the manual systems they replaced (more timely and economical as well). Each new application must pass this same test. As was indicated earlier, our initial model cost us under $3000 which included computer costs, systems analysis support and the time of our own staff.

From the models inception, other functional areas were asking us to change input variables and simulate the effects. We were immediately requested to incorporate additional features to enhance and generalize the use of the model for them.

There are several important contributing factors to this happy state. We have already discussed three important ones - usefullness, economy, and timeliness. There are other significant factors:

Interactiveness-the ability to change assumptions and obtain instantaneous results. The terminal and models essentially become an extension of the manager or analyst.

Direct Involvement -There are no third parties or intermediary obstacles to work through (coding sheets, keypunch routines or programmers), just the manager and the terminal.

APL Language -APL is extremely macro oriented and knowledge of only a few commands allows the user considerable "programming" capability. Our Management Sciences group (in the U.S. have found APL to save resources over conventional languages (Fortran, COBOL, BASIC) of from 5-15: 1 (Redwood 1972 and Schengilli 1971).

Timeliness -To implement useful models. The combination of terminals, on-line data bases and APL allowed models to be developed and implemented in the time that it would take to write the specifications in a more conventioanl language on a batch system.

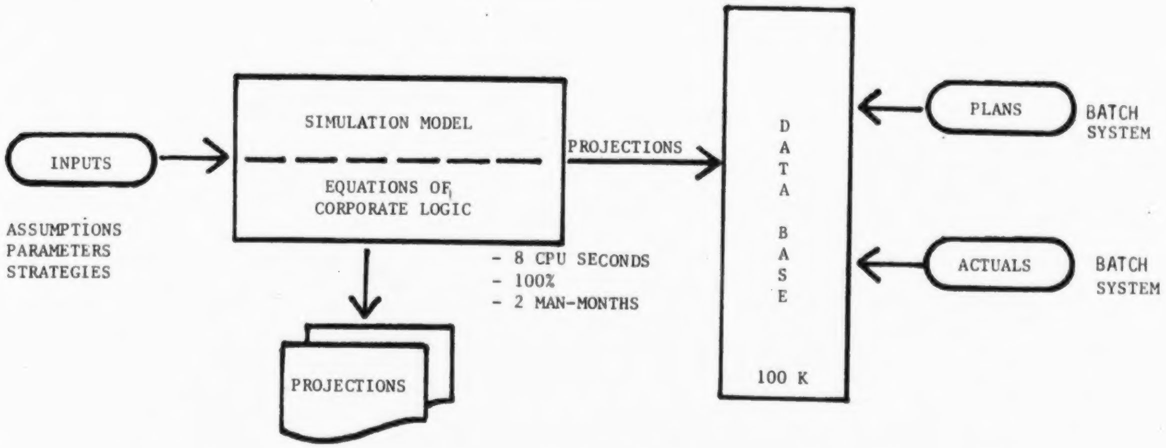
User Involvement -Because of all the above, it put a modelling capability directly in hands of the user - given that he has analytical ability or orientation.

All of the above models have been developed by my staff complemented with one full time consultant from the timesharing vendor and myself - a total of five. None of us have worked full time on modelling, the models have evolved as part of performing our overall responsibility. We estimate our total investment in these systems to be under $30K to date.

We have worked sufficiently long with these kinds of modelling applications to observe their effects on two other important work related aspects resource requirements and job satisfaction. We have saved considerable cost by using computer based models rather than strictly manual approaches. We estimate it would require several magnitudes of manpower increases to achieve the same level of output - a significant increase in efficiency. Each analyst has a low-cost terminal which they have come to rely upon in all phases of analysis and problem solving. The average usage is 50 hours per month per analyst. The range of APL programming capability ranges from poor to very good. However, to operate the models no programming ability is required, but I do require of the analysts, knowledge of how to access files and use the computer in a calculator mode. The use of timeshared models gives each of the analysts greater responsibility and scope than would be possible in a manual environment which has generally resulted in a much greater level of job satisfaction.

In conclusion we have found timeshared simulation models to be imminently successful in our planning processes. We use them as an augment to the manager in assisting him in planning and decision processes by providing the capability of an instantaneous data retriever, analyzer and projector. The models have been adopted in the U.S. as well, further attesting to their usefullness in these processes (Redwood) plus other uses. We have as well adopted the simulation models to our smallest divisions which demonstrates that computer based simulation models are economically viable in small as well as large business.

INITIAL SYSTEM


INPUTS
ASSUMPTIONS
PARAMETERS
STRATEGIES
SIMULATION MODEL
EQUATIONS OF CORPORATE LOGIC
PROJECTIONS
- 8 CPU SECONDS
- 100%
- 2 MAN-MONTHS
PROJECTIONS
D A T A B A S E
100 K
PLANS
BATCH SYSTEM
ACTUALS
BATCH SYSTEM

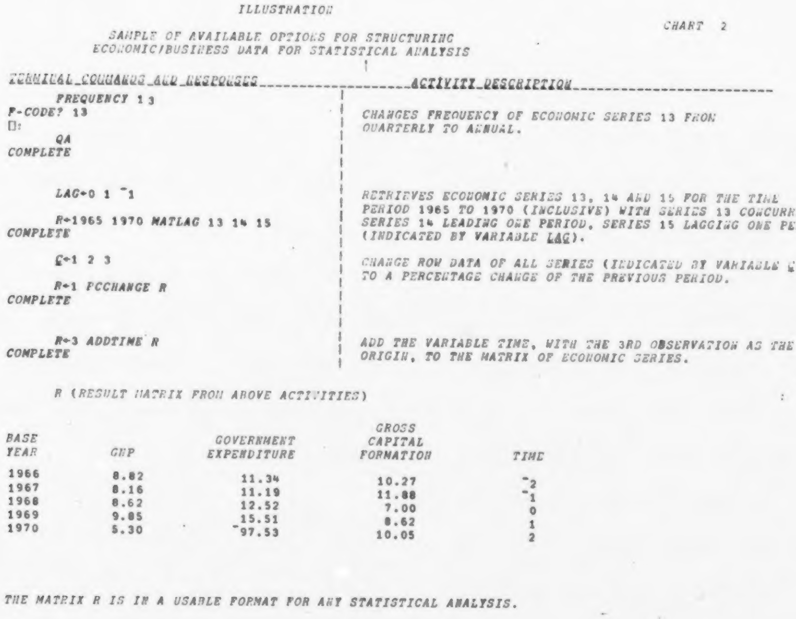
SIMULATION INPUTS (N30)

GROWTH RATES
TRADE ASSUMPTIONS
INVENTORIES - BEGINNING
COMPENSATION STRATEGIES
SEASONALITY PATTERNS

SIMULATION OUTPUT (N300)

ORDERS BY PRODUCT LINE
INVENTORIES
PERFORMANCE DATA
RESOURCE REQUIREMENTS
PROFORMA FINANCIAL STATEMENTS

CHART 1



```
                         ILLUSTRATION
                                                                   CHART  2
               SAMPLE OF AVAILABLE OPTIONS FOR STRUCTURING
             ECONOMIC/BUSINESS DATA FOR STATISTICAL ANALYSIS

TERMINAL COMMANDS AND RESPONSES                 ACTIVITY DESCRIPTION
      FREQUENCY 1 3                         |
F-CODE? 13                                  | CHANGES FREQUENCY OF ECONOMIC SERIES 13 FROM
⎕:                                          | QUARTERLY TO ANNUAL.
      QA                                    |
COMPLETE                                    |
                                            |
                                            |
      LAG←0 1 ¯1                            | RETRIEVES ECONOMIC SERIES 13, 14 AND 15 FOR THE TIME
                                            | PERIOD 1965 TO 1970 (INCLUSIVE) WITH SERIES 13 CONCURRENT,
      R←1965 1970 MATLAG 13 14 15           | SERIES 14 LEADING ONE PERIOD, SERIES 15 LAGGING ONE PERIOD
COMPLETE                                    | (INDICATED BY VARIABLE LAG).
                                            |
      C←1 2 3                               | CHANGE ROW DATA OF ALL SERIES (INDICATED BY VARIABLE C)
                                            | TO A PERCENTAGE CHANGE OF THE PREVIOUS PERIOD.
      R←1 PCCHANGE R                        |
COMPLETE                                    |
                                            |
                                            |
      R←3 ADDTIME R                         | ADD THE VARIABLE TIME, WITH THE 3RD OBSERVATION AS THE
COMPLETE                                    | ORIGIN, TO THE MATRIX OF ECONOMIC SERIES.

      R (RESULT MATRIX FROM ABOVE ACTIVITIES)

                                           GROSS
BASE                      GOVERNMENT      CAPITAL
YEAR         GNP          EXPENDITURE     FORMATION        TIME

1966         8.82            11.34          10.27           ¯2
1967         8.16            11.19          11.88           ¯1
1968         8.62            12.52           7.00            0
1969         9.85            15.51           8.62            1
1970         5.30           ¯97.53          10.05            2


THE MATRIX R IS IN A USABLE FORMAT FOR ANY STATISTICAL ANALYSIS.
```

CHART 2

```
                         ILLUSTRATION

          MULTIPLE REGRESSION OF FIRM DATA WITH ECONOMIC SERIES
   ANNUAL               )
   STARTYEAR 1959       )
   PREDICT ALL          )
   CLIM 0               )   INPUT
   SIGNIFICANCE 1       )   VARIABLES
   PITEST 3             )

   DV REGRESS IV  -  PROGRAM EXECUTION


MULTIPLE REGRESSION OF SALES AGAINST GNP AND CORPORATE PROFITS

COEFFICIENTS      ST. ERROR    T VALUES     VARIABLES

A       ¯84.659                             DV          1    SALES
B1        0.049       0.001      52.083     IV      40012    GNP
B2       ¯0.143       0.011     ¯12.573     IV      30867    CORPORATE PROFITS

STATISTICS:

R2        0.999     R        1.000      K          0.025
R2        0.999     R        1.000      K          0.027
SE       15.006     SE      15.916      DW         2.180
F      5554.629     DF       2.000      DF         7.000

PREDICTIONS:

                                           RESIDUAL
               ACTUAL     FORECAST       AMOUNT     PERCENT
1959           1233.0       1242.6         ¯9.6        ¯0.8
1960           1344.0       1326.4         17.6         1.3
1961           1408.0       1411.1         ¯3.1        ¯0.2
1962           1497.0       1484.0         13.0         0.9
1963           1645.0       1658.0        ¯13.0        ¯0.8
1964           1909.0       1928.4        ¯19.4        ¯1.0
1965           2149.0       2158.7         ¯9.7        ¯0.4
1966           2368.0       2340.0         28.0         1.2
1967           2633.0       2632.0          1.0         0.0
1968           2986.0       2990.9         ¯4.9        ¯0.2

 AVERAGE (ABOVE)                           11.9         0.7
 OVERALL AVERAGE                           11.9         0.7

 PREDICTION INTERVAL TEST

1969           3354.0       3201.5        152.5         4.8
1970           3650.0       3468.3        181.7         5.2
1971           4035.0       3802.6        232.4         6.1

  AVERAGE (ABOVE)                         188.9         5.4


NOTE: ACTUAL & FORECAST DATA ARE HYPOTHETICAL DATA
```

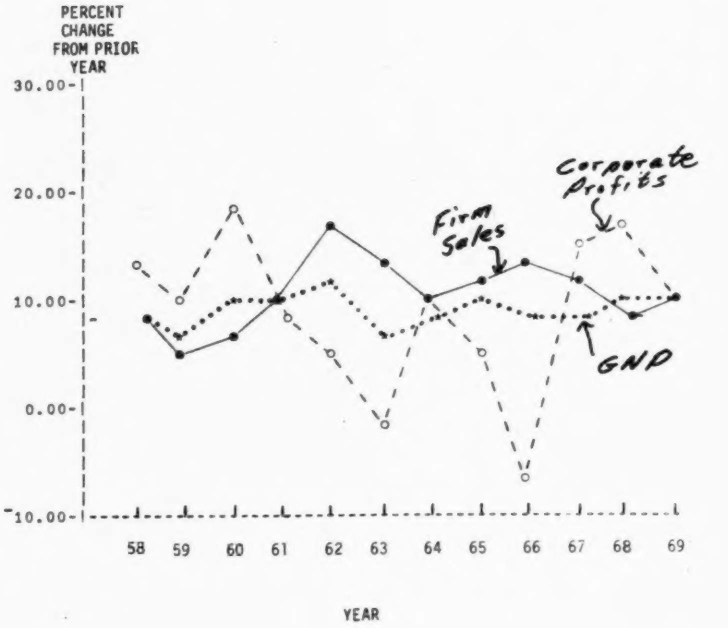


CHART 3

```
                                        AN ILLUSTRATION

                           FORECASTING WITH EXPONENTIAL SMOOTHING


MINIMIZING ESTIMATES-WEIGHTS BY P/L
NO/MAN

            MODEL         Dα          Tα          Sα        DEMAND      TREND       MAD

              1           .2          .5          .5          68        ¯2.00       13.8
              2           .2          .2                      79        ¯3.00       11.3
              3           .1                                  71        69.00       11.5


ORDERS
JUL OUTLOOK

         ------------------PERFORMANCE--------------------    -----OUTLOOK-----    ---------ACTUAL-OUTLOOK---------
                     -------JUN-------    -------YTD-------
  MODEL     FI       ACT    OUT    MAD     ACT    OUT    MAD   JUL    AUG    SEP     Q1     Q2     Q3     Q4    FYPC

    1      2.74      144    155     11     888    883     42   146    152    162     419    469    460    334  121.36
    2*     3.89      144    153      9     888    877     30   148    147    146     419    469    441    334  119.99
    3      3.70      144    173     29     888    911     35   150    149    148     419    469    447    334  120.42

MODEL--1=SEASONAL
       2=SINGLE
       3=DOUBLE
       *=OPTIMUM

                      α  =  OPTIMUM ALPHA (TIME) WEIGHTING FACTOR
                      D  =  DEMAND COMPONENT
                      T  =  TREND COMPONENT
                      S  =  SEASONALITY COMPONENT
                    MAD  =  MEAN ABSOLUTE DEVIATION
```

CHART 4

ILLUSTRATION

USE OF CURVE FIT PROGRAMS TO DEVELOP DATA FOR ECONOMIC FORECAST

```
         GPP CURVEFIT TIME


     GROSS PROVINCIAL PRODUCT FOR ONTARIO (1957 - 1969)

                          LINEAR               PARABOLA                CUBIC                QUARTIC              EXPONENTIAL
                   -----------------    -----------------    -----------------    -----------------    -----------------
            ACTUAL  FORECAST    DELTA   FORECAST    DELTA    FORECAST    DELTA    FORECAST    DELTA    FORECAST    DELTA

1957        13784    11134      23.8     13846      ¯0.5      13973      ¯1.4      13776       0.1      12566        9.7
1958        14060    12701      10.7     14057       0.0      14057       0.0      14189      ¯0.9      13544        3.8
1959        14829    14268       3.9     14515       2.2      14446       2.7      14637       1.3      14599        1.6
1960        15300    15836      ¯3.4     15219       0.5      15127       1.1      15234       0.4      15735       ¯2.8
1961        16010    17403      ¯8.0     16170      ¯1.0      16089      ¯0.5      16067      ¯0.4      16960       ¯5.6
1962        17016    18970     ¯10.3     17367      ¯2.0      17321      ¯1.8      17194      ¯1.0      18280       ¯6.9
1963        18699    20537      ¯9.0     18811      ¯0.6      18811      ¯0.6      18644       0.3      19703       ¯5.1
1964        20303    22104      ¯8.1     20501      ¯1.0      20548      ¯1.2      20420      ¯0.6      21237       ¯4.4
1965        22477    23671      ¯5.0     22439       0.2      22519      ¯0.2      22497      ¯0.1      22890       ¯1.8
1966        25342    25239       0.4     24622       2.9      24714       2.5      24822       2.1      24672        2.7
1967        27103    26806       1.1     27052       0.2      27122      ¯0.1      27313      ¯0.8      26592        1.9
1968        29566    28373       4.2     29729      ¯0.5      29729      ¯0.5      29861      ¯1.0      28662        3.2
1969        32493    29940       8.5     32653      ¯0.5      32526      ¯0.1      32329       0.5      30893        5.2

     AVG ERROR                   7.4                 0.9                 1.0                 0.7                  4.2

     COEFFICIENTS                 A            B            C            D            E
     ============
     LINEAR                   11134.0       1567.2
     PARABOLA                 13846.4         87.7        123.3
     CUBIC                    13973.3        ¯71.9        157.9         ¯1.9
     QUARTIC                  13776.2        439.9        ¯52.0         26.0         ¯1.2
     EXPONENTIAL                  4.1        NS

     ORIGIN:      OBSERVATION - 1


     NS=NOT SIGNIFICANT
```

CHART 5

CHART 6

ROLE OF MODELS IN THE CONTROL FUNCTION

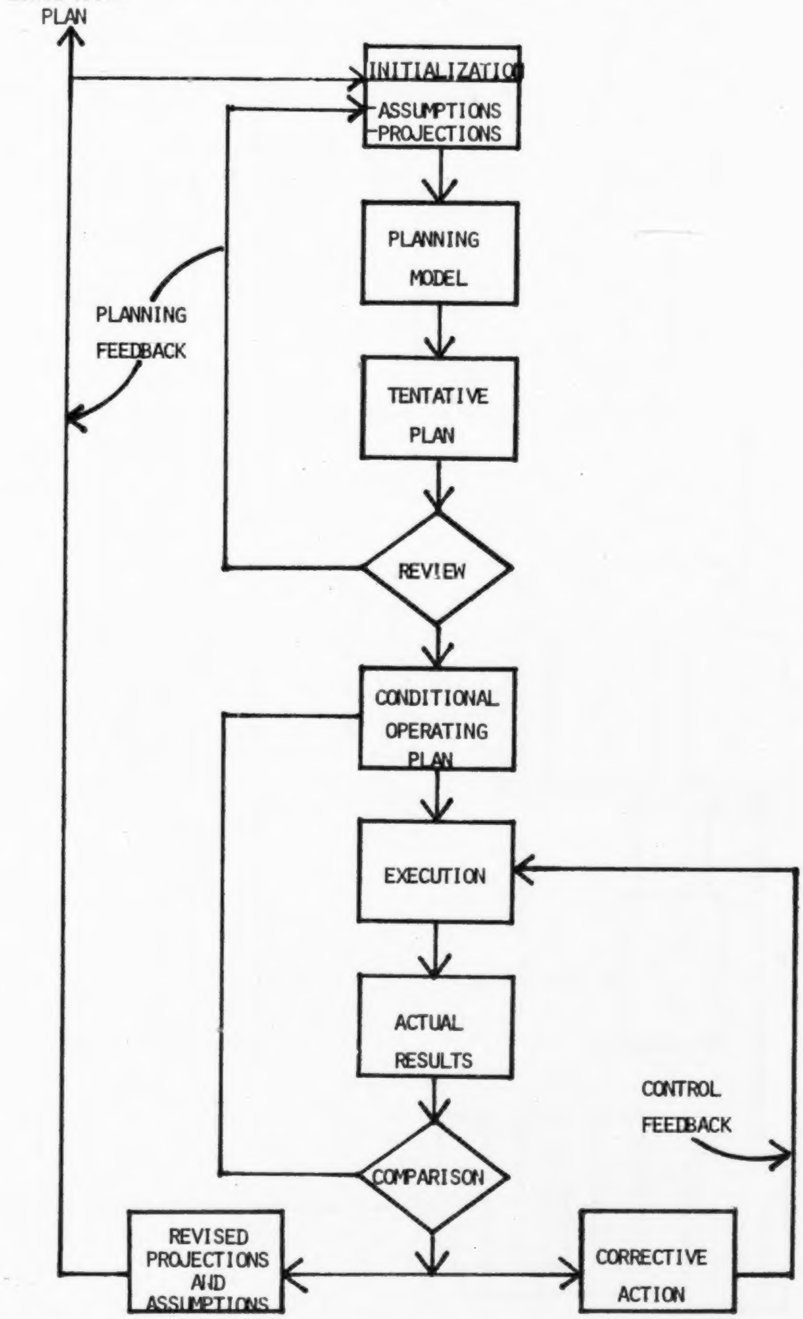
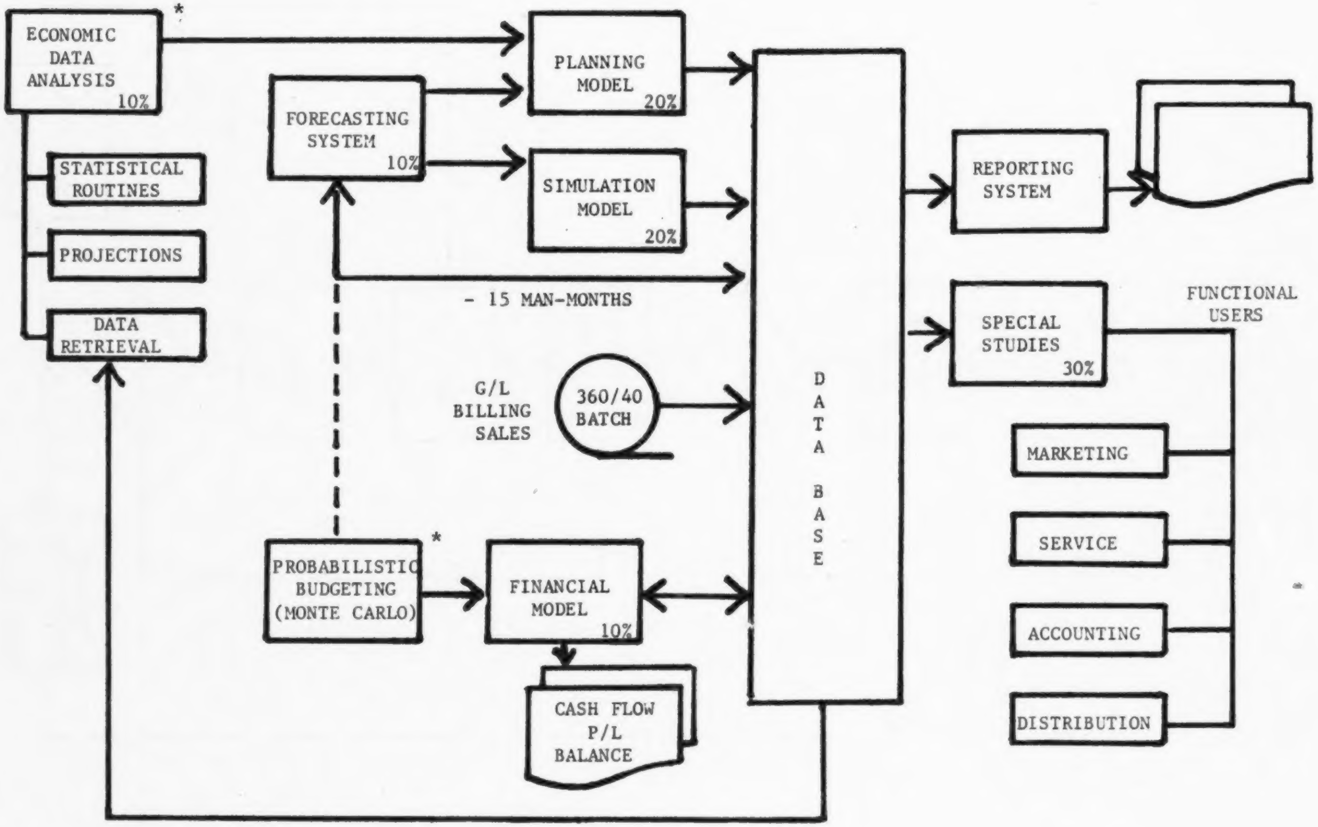

LONGE RANGE PLAN
INITIALIZATION
ASSUMPTIONS
PROJECTIONS
PLANNING MODEL
PLANNING FEEDBACK
TENTATIVE PLAN
REVIEW
CONDITIONAL OPERATING PLAN
EXECUTION
ACTUAL RESULTS
CONTROL FEEDBACK
COMPARISON
REVISED PROJECTIONS AND ASSUMPTIONS
CORRECTIVE ACTION

CHART 6

CHART 7

SYSTEM AT 28 MONTHS


ECONOMIC DATA ANALYSIS 10% *
STATISTICAL ROUTINES
PROJECTIONS
DATA RETRIEVAL
FORECASTING SYSTEM 10%
PLANNING MODEL 20%
SIMULATION MODEL 20%
- 15 MAN-MONTHS
G/L
BILLING
SALES
360/40 BATCH
PROBABILISTIC BUDGETING (MONTE CARLO) *
FINANCIAL MODEL 10%
CASH FLOW
P/L
BALANCE
D A T A B A S E
REPORTING SYSTEM
SPECIAL STUDIES 30%
FUNCTIONAL USERS
MARKETING
SERVICE
ACCOUNTING
DISTRIBUTION

* UNDER DEVELOPMENT

CHART 7

BIBLIOGRAPHY

Ackoff, R.L. *A Concept of Corporate Planning*, Wiley - Interscience, 1970.

Gershefski, George "Whats Happening in the World of Corporate Models", *TIMS Interfaces*, June 1971, p43.

Little, John D. "Models and Managers: Concept of a Decision Calculus", *Management Science*, April 1970.

Maxim D.L., Cook F.X. *Financial Risk Analysis*, American Management Association, 1972.

Redwood, Peter "APL for Business Applications", *Datamation*, May 1972.

Schengili, J. "So What's So Great About APL", *XEROX Data Processing Newsletter*, December 1971.

Schweyer H.E. *Analytical Models for Managerial and Engineering Economics*, Reinhold, 1964.

Seaberg, Charlotte "Business Applications in APL", Presentation to APL Conference at York University (Toronto), January 1972.

Seaberg, Charlotte "APL in Business", *Canadian Datasystems*, January-February 1971.

"Regional Models Come Into Their Own", *Business Week*, January 22, 1972, p72.

Richard D. Cuthbert
Manager, Operations Research
Corporate Information Services Division
Larry Peckham
Logistics Planning Specialist
Information Systems Group
Xerox Corporation, Rochester, New York

APL MODELS FOR OPERATIONAL PLANNING OF SHIPMENT ROUTING, LOADING, AND SCHEDULING

ABSTRACT

A vehicle routing algorithm with an imbedded loading heuristic and movement simulation was developed to provide the interactive capability to plan shipment delivery and returns.

The vehicle movement model is a small discrete event fixed interval simulation that uses the APL operators to keep track of loading facility status, truck location, etc. for scheduling purposes.

The models operate in a field environment, but are linked to centralized planning models and data bases.

They are illustrative of APL's almost unique ability to combine simulation modeling with operational restrictions such as interactive operation by non-programmers.

INTRODUCTION

This paper outlines the design of a system of APL models which support Xerox' distribution planning. It starts by describing the operating environment in which the planning occurs. This sets the stage for a discussion of the models' overall design philosophy: The power of simulation should be in the hands of the functional user rather than an operations research specialist. Some details are then given on the routing, loading, and scheduling sub-models to give insight into how they work together to aid development of a viable product delivery plan. This is followed by an explanation of how the truck scheduling simulation is conceived in terms of matrix (APL) operations. The way in which models are conceived in APL leads to a closing analysis of its advantages and disadvantages as a simulation tool, at least as experienced in this system.

SYSTEM OPERATING ENVIRONMENT

Exhibit 1 is a schematic of the system's operating environment. A national distribution staff is located in Rochester, New York. Regional operating staffs are located in five major cities: Chicago, Los Angeles, New York, Dallas, and Washington. The national staff is responsible for development of operating policies, such as what modes of transport can be used. In line with these policies, they negotiate with carriers and develop data bases needed to plan operations, such as freight rates by transport modes. Responsibility for regional movement of the product (Xerox copiers) is in the hands of the regional staffs who must also supply data on how well they are performing to Rochester.

The regional staffs must also develop the actual plans to move the machines in time to notify the carriers. They normally do this once a week and the calculations must be completed in the span of a few hours. They obtain requests for copiers from branches and must decide:

- *How many trucks* from each kind of carrier to use.
- What *routes* should the trucks take.
- How should the trucks be *loaded*.
- When should the trucks be *scheduled* to depart and arrive.

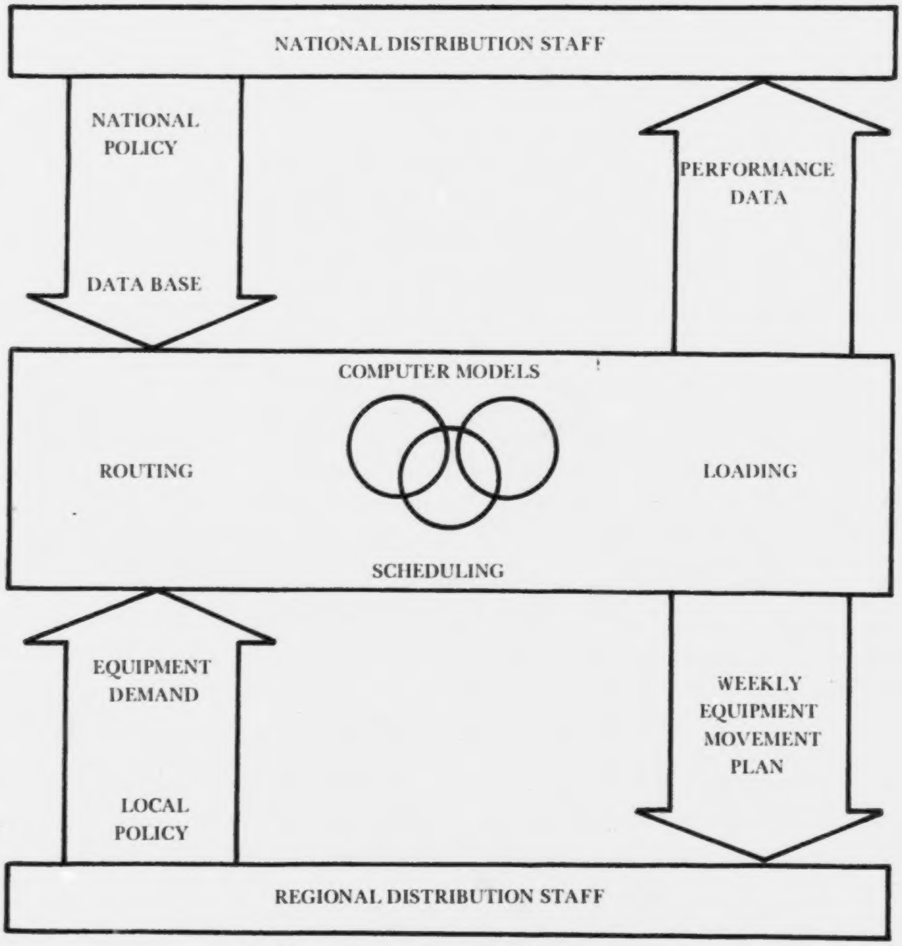

NATIONAL DISTRIBUTION STAFF
NATIONAL POLICY
DATA BASE
PERFORMANCE DATA
COMPUTER MODELS
ROUTING
LOADING
SCHEDULING
EQUIPMENT DEMAND
LOCAL POLICY
WEEKLY EQUIPMENT MOVEMENT PLAN
REGIONAL DISTRIBUTION STAFF

Exhibit 1: System Operating Environment

These decisions must take into account local policies and conditions known only to regional personnel; e.g., a temporary shortage of trucks available to the preferred carrier. The decisions are also interlocked; e.g., how many trucks are needed depends on how they are loaded, which in turn depends on how they are routed, and so on.

It is clear that the typical off-line optimization study by an operations analyst has no place in this environment. What is needed is a system of models that:

- Can be used by non-specialist personnel.
- Has very quick turnaround.
- Is very amenable to operator intervention to reflect local conditions.
- Can be fed data by a central staff of functional experts.
- Can be maintained by a central staff of operations analysts.

OVERALL DESIGN

Exhibit 2 shows the overall design of the system. This design is aimed at answering the needs of the distribution planners just discussed through exploitation of the power of modeling in general and of APL in particular. Later we shall review APL as a vehicle for simulation in the light of this design experience.

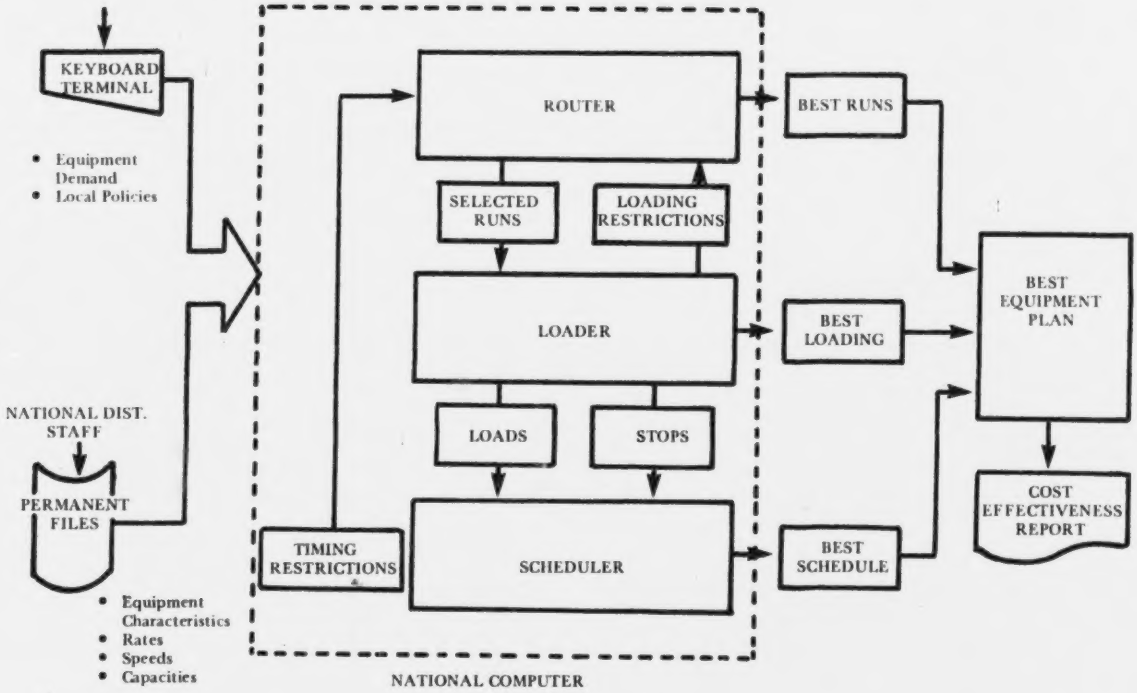

KEYBOARD TERMINAL
• Equipment Demand
• Local Policies
NATIONAL DIST. STAFF
PERMANENT FILES
• Equipment Characteristics
• Rates
• Speeds
• Capacities
ROUTER
SELECTED RUNS
LOADING RESTRICTIONS
LOADER
LOADS
STOPS
TIMING RESTRICTIONS
SCHEDULER
NATIONAL COMPUTER
BEST RUNS
BEST LOADING
BEST SCHEDULE
BEST EQUIPMENT PLAN
COST EFFECTIVENESS REPORT

Exhibit 2: Overall Systems Design

The first feature of the planning system that should be noted is that the computer, the models and all data files are centrally located. The work was initially done on an IBM 370-145 owned by an outside time sharing service. It is currently being converted to our own XDS Sigma-7 in Rochester (an APL interpreter has only recently become available on that machine). Central location enables us to keep national level control over model refinements and data base updates. All interfacing with model including data entry is done by teletypes over phone lines, so actual physical location of the computer is immaterial except for phone charges.

The national distribution staff maintains files on product characteristics, such as shipping weight, freight rates, vehicle capacities, speeds, etc. They enter this data via file maintenance procedures over a teletype. The data is stored permanently on disk packs at the computer center. Updating is done on an as needed basis.

The regional distribution staff enters weekly branch demand for copiers, and also through answering questions, determines the local operating policies under which planning is done.

The local planner seeks to develop a best equipment plan composed of best (near cost optimum) runs or routes, best (highest feasible load factor) loading, and best (minimum vehicle usage) truck schedule.

In this task he is aided by three interlocking sub-models: a router, a loader, and a scheduler. The router examines a file of available routes and queries the loader as to how many trucks would be required to meet input demand. The corresponding delivery cost is also calculated. The router selects the cheapest routes and transport modes and ignores scheduling problems. The interaction between the router and loader is entirely automatic; i.e., occurs without operator intervention. What is produced is the cheapest routing consistent with truck capacities, but which may be infeasible from a scheduling viewpoint.

The human operator then intervenes to determine if the cheapest routing is feasible from a scheduling viewpoint. He inputs departure times for the cheapest runs consistent with his dock loading capacity. A truck movement simulator traces the movement of each vehicle and informs him of all their activities. If the trucks' arrival times are not satisfactory, he can manually input new departure times until they are; then his job is done.

If, however, no departure times can be found to make the cheapest runs feasible, he may instruct the routing routine to modify its recommendations; e.g., restrict use of one or more runs.

This iterative process is repeated until the operator is convinced he has the cheapest route consistent with all operating constraints even those the model is not directly aware of. In other words he conditions the computer solution to make it feasible in light of non-quantifiable constraints.

In all cases the model informs the operator of the cost of all alternatives. If, for example, a branch requests crash service not within the optimal schedule, he can evaluate the cost and make a decision whether to approve the variance from plan.

This segmented design approach was taken because it represented a good mix of computer and human talents under the technical limitations of APL. The calculation burden of searching thousands of routing alternatives was taken off the shoulders of the planner. The human supplied full recognition of rapidly changing local conditions. How much the technical limitations of APL influenced the design tradeoff is a later topic of discussion.

TRUCK ROUTING, LOADING, AND SCHEDULING CONCEPTS

Exhibits 3, 4, and 5 show in conceptual terms how truck routing, loading, and scheduling are accomplished.

The process begins with the planner entering the weekly demand for product pickups and deliveries through a 'START' routine. After this data is edited and filed, the START routine calls the ROUTE routine to determine a good set of truck runs. The planner can force the computer

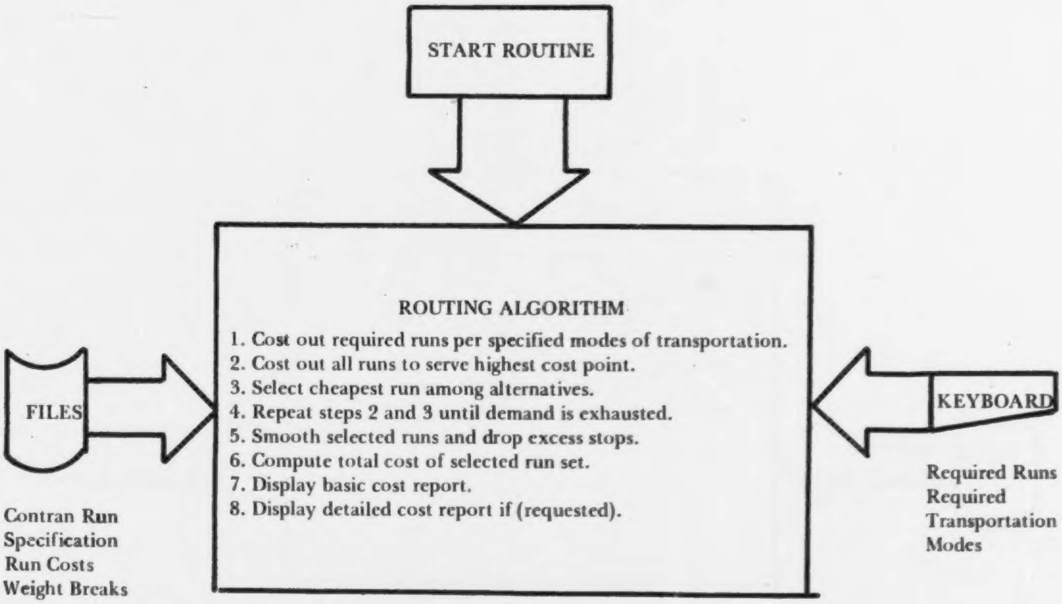

START ROUTINE
ROUTING ALGORITHM
1. Cost out required runs per specified modes of transportation.
2. Cost out all runs to serve highest cost point.
3. Select cheapest run among alternatives.
4. Repeat steps 2 and 3 until demand is exhausted.
5. Smooth selected runs and drop excess stops.
6. Compute total cost of selected run set.
7. Display basic cost report.
8. Display detailed cost report if (requested).
FILES
Contran Run
Specification
Run Costs
Weight Breaks
KEYBOARD
Required Runs
Required
Transportation
Modes

Exhibit 3: Truck Routing Concept

to make some runs for operating reasons and these will be routed and loaded first. The ROUTE routine depends on the LOAD routine to tell it how many trucks are needed and what is their load factor.

After all required runs are completed, the computer examines all branches and determines which is most expensive to serve with direct (1 stop) service. It then searches all filed multistop runs (CONTRAN) that go to that point and selects the cheapest. The machine demand is decremented by the truck's load and the process is repeated until all demand at all points is exhausted.

Then, because the above heuristic is "greedy," loads are shifted among the chosen runs to smooth the loading. This may eliminate some excess charges incurred for very heavy loads and also some unnecessary stops.

The ROUTE routine also displays the routing's total cost and run costs if desired.

The loading concept as in Exhibit 4 is interesting in that it deals with products that are relatively large compared with the vehicle. For example, only three of our large machines can be fit side by side on a truck. Products of different sizes are normally shipped together. This means space can be wasted if the loading pattern is not well laid out.

The routing logic gives the loader data on the machine demand at the potential stopping points. The loader loads the stops in reverse order since there are doors only at the end of the truck.

Big machines (duplicators) are loaded first since they are the hardest to fit. They are loaded in from front to back until they are exhausted. Then middle size machines, known to fit, are filled in beside them. Next, consoles and desktops are put in as sets of 3, 4, 5 until the first deck of the truck is exhausted. A second deck is then gone to until the truck's overall capacity is exhausted. Legal weight limits are also observed.

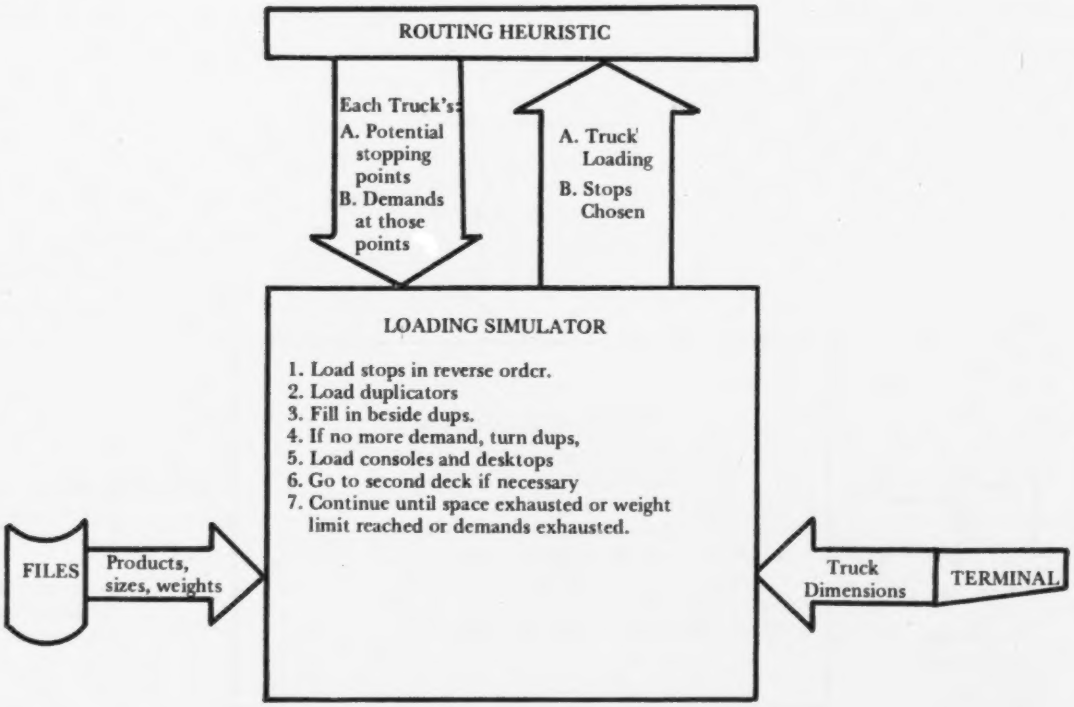

ROUTING HEURISTIC
Each Truck's:
A. Potential stopping points
B. Demands at those points
A. Truck Loading
B. Stops Chosen
LOADING SIMULATOR
1. Load stops in reverse order.
2. Load duplicators
3. Fill in beside dups.
4. If no more demand, turn dups,
5. Load consoles and desktops
6. Go to second deck if necessary
7. Continue until space exhausted or weight limit reached or demands exhausted.
FILES
Products, sizes, weights
Truck Dimensions
TERMINAL

Exhibit 4: Truck Loading Concept

This loading process is essentially a simulation of human dock loading practice and was tested and refined through competition with experienced dock personnel.

Note, however, that the point is not to "beat" the man on the dock, but to make the planner aware of how the dock man would behave if he were told to load a given block of machines. This enables the planner/computer to try many different routings and loadings to determine the cheapest.

Exhibit 5 shows the scheduling concept and how it relates to the rest of the system.

The scheduler is called upon to determine the truck departure and arrival times for the runs initially selected by the routing process.

The scheduler accepts manual starting times and simulates truck movements taking into account truck speeds, inter-city mileages, legal restrictions on driving hours, and the operating schedules of the delivery point facilities. How this simulation works in detail is discussed in a moment to give a better feel for APL model formulation.

The computed arrival times are displayed so that the planner can determine if the schedule is satisfactory. If it is, the planning job is completed. If it is not, he can juggle departure times to avoid dock congestion or weekend runs until he arrives at a satisfactory schedule. Occasionally he is forced to redo the routing process with restrictions on the use of runs that cannot be satisfactorily scheduled. In other words, the routing-scheduling process is a feedback iterative process controlled by the systems operator.

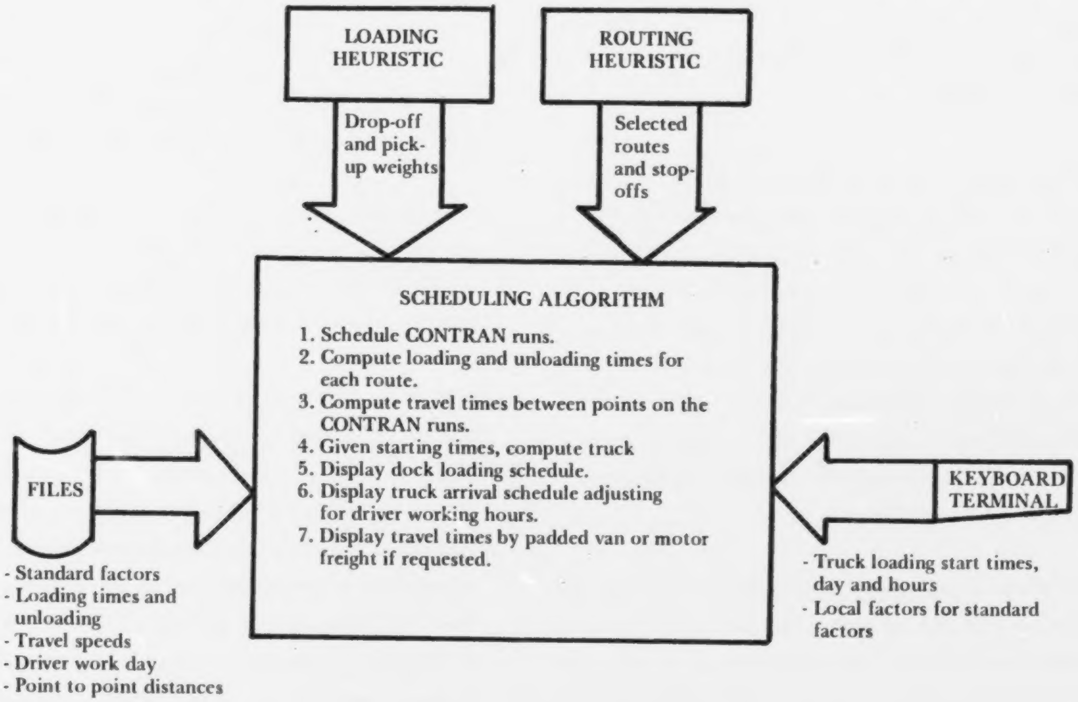

LOADING HEURISTIC
ROUTING HEURISTIC
Drop-off and pick-up weights
Selected routes and stop-offs
SCHEDULING ALGORITHM
1. Schedule CONTRAN runs.
2. Compute loading and unloading times for each route.
3. Compute travel times between points on the CONTRAN runs.
4. Given starting times, compute truck
5. Display dock loading schedule.
6. Display truck arrival schedule adjusting for driver working hours.
7. Display travel times by padded van or motor freight if requested.
FILES
- Standard factors
- Loading times and unloading
- Travel speeds
- Driver work day
- Point to point distances
KEYBOARD TERMINAL
- Truck loading start times, day and hours
- Local factors for standard factors

Exhibit 5: Truck Scheduling Process

APL FORMULATION OF THE TRUCK MOVEMENT SIMULATION

The SCHEDULE routine is required to determine the arrival time of trucks given departure times, truck speeds, inter-city distances, etc.

Exhibit 6 shows how this problem was formulated for APL coding. First, each trip was conceived of as occurring without delays due to facility shutdowns or logged driving hour limitations. The truck was loaded, rolled to the first stop, unloaded, rolled to the next stop, etc.

Inter-city distances and driving speeds were drawn from files and travel times were computed from their ratios. Shipment weight at each stop was drawn from the routing logic and multiplied by a loading factor per pound. The sum of the loading time at the prior stop and the inter-city travel time between each stop and the prior stop, gives the time between arrivals in working hours.

Input supplies working and driving hour limits and facility open hours. This information is used to set up facility and crew status vectors.

A time counter is advanced starting at the input beginning point. The truck is loaded for as many hours as it takes to fill it. The crew's time is not taken up in the initial loading. Once the truck is full, the crew's status is marked as working and driving. This status is kept as each hour passes until an arrival is due. The crew's log book or status vector is scanned each hour to determine how long they have worked. When they have worked their limit, they are forced to sleep. If the truck arrives at a facility whose status is closed, time advances and the crew is forced to "sleep" again.

The effect of this process is to interject dead time into the arrival time schedule originally computed. In other words, we convert working hours into calendar hours by taking into account departure time, sleeping time, and time spent waiting at shut facilities. There is no way these dead times could be calculated without tracing the prior history of truck movement, hence the procedure is rightfully called a simulation.

The advantage of the approach is that in general individual entities and events need not be traced in the coding. For example, the array ARIV is a single matrix that is computed in a few steps without iterations or scanning to trace an individual truck making individual stops. Matrix operations (x ÷ etc.) permit doing the job in one pass.

Other operations are used to flag facilities as open or shut to all trucks. In other words, as much homework as possible is done before getting into the time scanning (which does involve a FORTRAN type loop). In addition, historical scans of the facility and crew status vectors are done directly with single operators. This prevents loops occurring within loops.

These steps are taken because:

- It is conceptually easier to use the APL primitive operators (after you get used to the idea).
- APL is interpretive and loops should be avoided since each line would be retranslated over and over.

When formulating APL problems for the experienced coder, it is often not necessary to go into much more detail than Exhibit 6. This enables the analyst and programmer to converse on the conceptual level rather than the coding level.

APL AS A SIMULATION MODELING TOOL

Exhibit 7 summarizes our experience with APL as a simulation modeling tool. In systems development, the main thing that stands out is the sheer coding speed of APL over FORTRAN. For heuristics, models, and simulations, we estimate it to be perhaps five times faster. This savings comes from the fact that it is possible, as we have seen with the truck movement simulator, to conceive the coding in terms of matrix operations that are conceptually very close to the basic processes to be modeled.

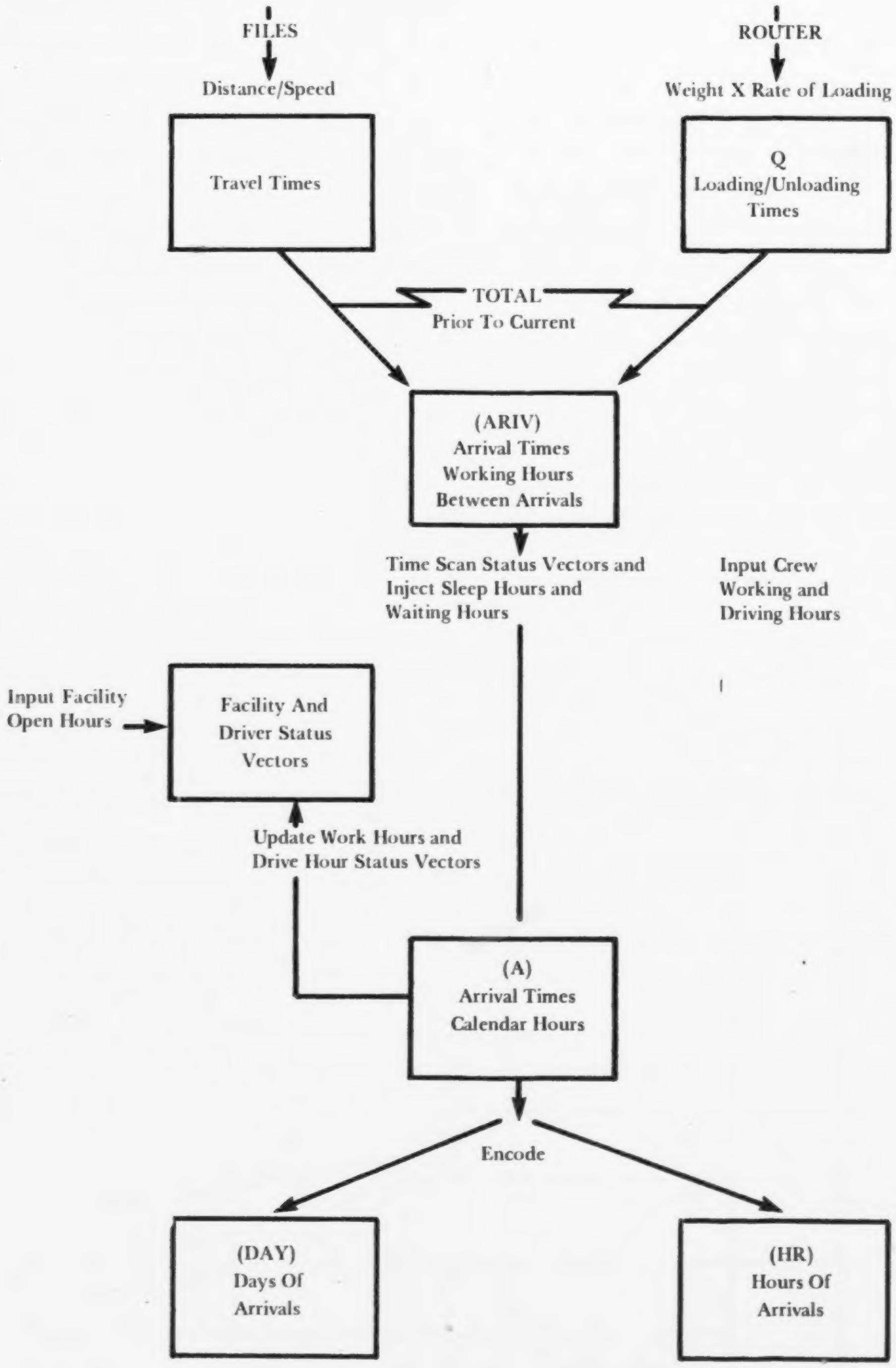

FILES
Distance/Speed
ROUTER
Weight X Rate of Loading
Travel Times
Q
Loading/Unloading
Times
TOTAL
Prior To Current
(ARIV)
Arrival Times
Working Hours
Between Arrivals
Time Scan Status Vectors and
Inject Sleep Hours and
Waiting Hours
Input Crew
Working and
Driving Hours
Input Facility
Open Hours
Facility And
Driver Status
Vectors
Update Work Hours and
Drive Hour Status Vectors
(A)
Arrival Times
Calendar Hours
Encode
(DAY)
Days Of
Arrivals
(HR)
Hours Of
Arrivals

Exhibit 6: Scheduling Simulation APL Formulation

Besides reducing the coding bottleneck, APL's speed has another major influence on the project. It made it possible to experiment with alternative heuristics on an ongoing basis; i.e., to conduct research as part of the planning system design. This turned out to be very important because we were working in an area where operations research was required before the final design could be selected. For example, we originally thought linear weight and area constraints would be sufficient to determine product loading feasibility. Experiments comparing the loads obtained against dock loading practice showed it was necessary to simulate the fitting of each machine on board the truck.

If coding speed is APL's strength, then its biggest weakness in our experience was the small available workspace. The total storage requirement of the system exceeded the workspace available, forcing us to segment the routing and scheduling into two separate workspaces. We don't believe that this hurt the project too badly in terms of performance of the system because the interface between these two processes probably will always require manual intervention anyway. However, under current APL software, program swapping or chaining is not possible at the speeds necessary for algorithmic searches or large scale simulations. This is a major limitation of the language which is currently being addressed by software

SYSTEM CHARACTERISTIC			INDICATOR OR EXPERIENCE		COMMENTS
			Elapsed Time (Months)	Effort (Man Months)	
Development	Design		1.0	2.0	Time largely unaffected by APL characteristics
	Coding		1.0	2.0	Major reduction in time over FORTRAN due to APL coding speed.
	Test And Implementation		3.0	9.0	Major reduction in time over FORTRAN due to rapid rewrite capability and terminal editing.
Operating	Time	CPU	3 - 4 Minutes		Much heavier CPU use not practical on interactive basis.
		Terminal Connect	20 - 40 Minutes		Much longer elapsed times not practical on interactive basis.
	Size	Coding	Almost 48,000 bytes or 6,000 characters		Coding is stored and interpreted - not compiled.
		Total	About 75,000 bytes		Small workspace size (48,000 bytes) forced use of program segmenting.
	Cost	CPU	$30		Estimated payoff per use: $200
		Terminal Connect	$5 - 10		Relatively small proportion of total cost for APL application. Depends on distance to computer.

Exhibit 7: Summary of System Characteristics

groups. Disk files for data storage are available and were used extensively.

In terms of running time, APL is faster than we originally anticipated, but slow enough to force us to limit the amount of route searching done in the system.

APL is an interpretive language; i.e., it retranslates source coding on a line-by-line basis. This imposes a translation overhead which can be very severe in highly repetitive search processes. It is not only desirable to conceive models in terms of matrix operations, it is mandatory if speed is a consideration. Carrying out processes as in FORTRAN row-by-row or column-by-column is not practical.

If the model is to operate in an interactive mode (as do most APL models), the amount of calculation that is practical is limited. In our system the user, at some points, must wait up to 10 minutes elapsed time for calculations to be complete. Phone connections are not reliable enough to push that figure much higher.

Fortunately we found that the system helped the planner reduce the distribution costs significantly within the limits imposed by the technology. The direct savings came out about 5-10 times system operating costs.

CLOSING SUMMARY

Summarizing our experience with APL in this system, we were generally impressed. We are converting most of our modeling efforts over to the language because of its coding speed advantages. We are also requesting our software groups to address the program chaining and workspace size issues on our own machine. We anticipate a 50% increase in workspace size.

Perhaps the best thing that can be said in APL's favor is that it makes it technically possible to put models where they belong (in the planner's hands) in an amazingly short period of time.

He then can participate in the testing process and give much more rapid feedback on an approach's relevance to the real problem. This goes a long way in overcoming the implementation gap that often occurs with operations research based projects.

A TWO ASSET CASH FLOW SIMULATION MODEL

Richard C. Grinold

and

Robert M. Oliver

University of California, Berkeley

Abstract

We present an *APL PLUS* simulation model of a two asset cash management system. The two assets are cash and an income producing portfolio. The model relates cash transfer policies and futures stocks of cash and income producing assets to restrictions on average and minimum balances, projections of future cash flows, the costs of cash management, and user specified objectives.

I. Introduction

An organization's cash management system acts as a buffer between the organization's internal cash needs and its external financial relations with customers, suppliers, tax collecting agencies, and banks. Three frequently cited reasons for holding cash are to ease anticipated disbursements, to act as a reserve for unexpected cash demands, and to compensate banks for their services. The most important cost of holding cash is the opportunity lost by not using the cash in some alternate way. For example, cash could be used to pay existing loans, as a dividend to shareholders, as a short term external investment, or as an internal capital investment. The opportunity cost for holding cash is the loss incurred by not using the cash in some optimal alternative manner.

Cash flow management involves scheduling the timing and size of cash transfers in order to meet the organization's needs without carrying excessive amounts of cash in nonproductive accounts. The two asset cash management model presented in this paper is based on a forecast of future cash inflows, institutional restrictions on cash stocks, the costs of holding and transferring cash, and a user selected objective. The model is not unnecessarily complicated and includes many essential aspects of the cash management problem. The simplicity and interactive nature of the computer model make it an ideal simulation tool for decision makers. With practice, the input and output of the model can be quickly analyzed. This interactive feature allows the decision maker to simulate the consequences of changes in cash flow forecasts, opportunity costs, compensating balance agreements, and cash management objectives.

Most large organizations have a choice of several income producing assets that are alternatives to holding cash. These assets differ in maturity, risk, and yield. The cash management model presented in this paper aggregates these income producing assets into a single income producing portfolio. The two assets, cash and the income producing portfolio, are similar to a household's checking and savings account. If a firm is constantly in a borrowing position, then the income producing asset can be viewed as a loan portfolio. The two asset assumption leads to considerable simplifcation, and can be considered as a first approximation to the multiple asset model. Cases in which the two asset assumption is inappropriate, will be discussed in the summary.

Similar cash management models have been considered in references (1-3). The models in (2-3) lead to large linear programs that are not suitable for flexible, interactive simulation. The model in (1), is based on the special structure of optimal policies in a dynamic program which includes fixed transaction costs but does not examine average balance constraints.

The next section describes the *APL* simulation package, outlines the reasons for using *APL* as a simulation language, and suggests several uses for the model. Section III formulates the cash management problem and describes several alternative cash management objectives. The fourth section contains an example of marginal analysis under the special objective of minimizing the total cost of cash management. A similar marginal analysis can be developed for alternative objectives. Section V presents some examples, and indicates how optimal policies change in response to changes in external system parameters. The sixth and final section summarizes the results.

II. *APL* Simulation

The simulation is based on an interactive system of *APL*/360 programs. These programs:

(a) Examine a set of globally defined variables.

(b) Construct a linear programming tableau appropriate to the length of the planning horizon.

(c) Insert coefficients that are computed from the global variables, (a), and the underlying cash flow conservation equations.

(d) Include user imposed constraints, such as compensating balances and/or a lower bound on the cash level.

(e) Calculate the optimal timing and size of cash transfers, and the marginal values of changes in exogenous cash flows, compensating balance levels and lower bounds on the cash level.

(f) Organize the output in a graphical or tabular display as specified by the user.

APL was selected as the program language for two reasons. First, the interactive aspect of the language allows users to employ the programs frequently and within the short timespan available for making cash management decisions. Second, the compact, array oriented mathematical structure of *APL* is ideally suited for constructing the appropriate tableau, for solving the optimization problem, and for quickly modifying the model's assumptions and parameters through user defined global variables.

The simulation model is built around a linear program that calculates optimal cash transfers and stock levels. The interactive computer program allows a decision maker to simulate the impact of changes in predicted cash flows, cash management objectives, institutional constraints, and the length of the planning horizon on cash transfer decisions and on future stocks of cash and the income producing asset. The program package can be used in several ways.

(a) To find feasible cash flow transfer schedules that will meet the organization's commitments. This scheduling problem can be considered in the long run (monthly periods), in the short run (daily periods), or over a sequence of unequal periods (days, weeks, months, quarters, years).

(b) To estimate the costs of cash management, and to project the future stocks of cash and the income producing asset.

(c) To determine the costs of institutional arrangements such as compensating or minimum balance agreements.

(d) To examine the impact of smoothing cash flows.

(e) To forecast periods in which short-term borrowing will reduce costs.

(f) To identify unforseen investment possibilities.

(g) To test the effect of alternate cash management objectives on cash management policy.

(h) To calculate the cash transfers and stock levels of cash and the income producing asset that result when the worst possible cash inflow occurs.

(i) To gauge the impact of alternate cash flow forecasts on costs and cash transfer decisions.

(j) To discover a suitable planning horizon for cash management planning.

III. Formulation

This section formulates the cash flow model and indicates some of the features that the simulation program can accommodate.

The fundamental cash flow conservation equation

$$(1) \qquad X[J] = X[J-1] + S[J] + U[J] - V[J] \quad \text{for } J = 1, 2, \ldots, T$$

with $X[0] = X\underline{Q}$ a user supplied global variable.

Period J is an interval of time from instant J-1 to instant J. The variables in (1) are defined as follows: $X[J-1]$ is the cash level at the end of period J-1 ; $U[J] - V[J]$ is the net amount of cash transferred into, $U[J] > 0$ for an inflow and $V[J] > 0$ for outgo, the cash account at the beginning of period J, (at time J-1). $S[J]$ is the net inflow of cash during period J and $X[J]$ is the cash level at the end of period J, (at time J). The initial cash level $X[0]$, and the exogeneous cash inflows $S[J]$ are known. If a lower bound determined by the global variable $L\underline{X}$, is placed on the cash account then equation (1) continues to hold although $X[J]$ in (1) must be interpreted as the cash level in excess of the lower bound; i.e. the actual cash level minus $L\underline{X}$. A compensating balance requirement is described by a constraint on the total cash holding.

$$(2) \quad X[1] + X[2] + \ldots + X[T] \geq (T \times (A\underline{X} - L\underline{X}))$$

Here $A\underline{X}$ is a global variable describing the average level in the cash account. The simulation program ignores the average balance constraint if $A\underline{X} < L\underline{X}$.

The cash management objective is selected through a user defined global variable $O\underline{F}$. The four possibilities are:

$O\underline{F} \leftarrow 1$: Minimum sum of all future opportunity and transfer costs

$\leftarrow 2$: Minimum discounted sum of all future cash transfers

$\leftarrow 3$: Maximum end of period cash and securities

$\leftarrow 4$: Maximum present worth of future cash and securities

Other objective functions can easily be handled. The model is able to simulate the policy implications of following any specified objective.

Other features of the package allow for

unequal planning periods, and allow for multiple average balance constraints. For example, the planning period could consist of eight weekly periods followed by four monthly periods. If the organization is constrained on an average monthly balance then there are a total of six balance constraints. The first involves $X[1]$ through $X[4]$, the second $X[5]$ through $X[8]$ and the last four simply state that $X[J] \geq A\underline{X}$ for $J \geq 9$. The complete list of user supplied global variables is described below.

$X\underline{O}$ - initial cash stocks

$Y\underline{O}$ - initial stock of income-producing assets

$C\underline{X}$ - variable unit "opportunity" cost of holding cash

$C\underline{U}$ - variable unit cost of transferring assets to cash

$C\underline{V}$ - variable unit cost of transferring cash to assets

$A\underline{X}$ - the average cash level or compensating balance

$L\underline{X}$ - lower bound for cash stocks

$O\underline{F}$ - an integer denoting the choice of objective function

$\underline{D}$ - discount factor

$\underline{R}$ - unit period interest rate

$D\underline{P}$ - a vector giving the number of days, weeks or months in each accounting period

$B\underline{P}$ - the number of periods that apply to a compensating balance restriction

To avoid complications we shall concentrate our analysis on the special case $D\underline{P} = 1$, $O\underline{F} = 1$, and $B\underline{P}$ equals the length of the planning horizon.

IV. Marginal Analysis

This section describes in detail the marginal cost information derived in the simulation when the cash management objective is the minimization of total opportunity and transfer cost. A similar analysis is possible for the other cash management objectives mentioned in Section III. The information derived in this section will be useful in interpreting the numerical examples that are presented in Section V.

The primal linear program minimizes the total opportunity and transfer cost subject to the flow equations (1) and the total balance restriction (2). The dual linear program associates variables $P[1], P[2], \ldots, P[T]$ with the flow equations and a variable Q with the balance condition. These variables must satisfy the dual feasibility conditions

$$(3)\quad \begin{array}{ll} \text{(i)} & -C\underline{U} \leq P[J] \leq C\underline{V} \qquad J = 1,2,\ldots,T \\ \text{(ii)} & P[J] - P[J+1] \leq C\underline{X} - Q \end{array}$$

Where we assign $P[T+1] = 0$.

In addition, a primal solution (X,U,V) and dual solution (P,Q) are optimal if and only if they satisfy the complementarity conditions

(i) $-C\underline{U} < P[J] < C\underline{V} \Rightarrow V[J] = U[J] = 0$

$$V[J] > 0 \Rightarrow P[J] = C\underline{V}$$

(4) $$U[J] > 0 \Rightarrow P[J] = -C\underline{U}$$

(ii) $P[J] - P[J+1] < C\underline{X} - Q \Rightarrow X[J] = 0$

$$X[J] > 0 \Rightarrow P[J] - P[J+1] = C\underline{X} - Q$$

The variable $P[J]$ is the rate of change of the minimal cost with respect to changes in $S[J]$. With this interpretation of P in mind it is easy to derive (3) and (4). Increase $S[J]$ by a small amount E . This change can be offset by a corresponding increase in $V[J]$. The increase in cost is $E \times C\underline{V}$ which must be larger than the increase in the minimal cost $E \times P[J]$. Since $E > 0$, it follows that $C\underline{V} \geq P[J]$. If, $V[J] > 0$ then the argument above could be repeated with E negative. Then $V[J] > 0$ could be decreased. The increase in cost remains $E \times C\underline{V} \geq E \times P[J]$. However, since $E < 0$, it follows that $C\underline{V} \leq P[J]$. Therefore $V[J] > 0$ implies $C\underline{V} = P[J]$.

A similar argument holds for $U[J]$. If $S[J]$ decreases by $E > 0$ then this decrease can be offset by a corresponding increase of E in $U[J]$. The increase in cost is $E \times C\underline{U} \geq -E \times P[J]$. Therefore $C\underline{U} \geq -P[J]$. However when $U[J] > 0$, the same argument can be repeated with $E < 0$. It follows that $U[J] > 0$ implies $C\underline{U} = -P[J]$.

The variable Q measures the rate of increase in optimal cost per unit of increase in the total balance constraint (2). Suppose, that $S[J]$ is increased by E , $S[J+1]$ is decreased by E , and the total balance requirement is increased by E . If $E > 0$ this change can be offset by increasing $X[J]$ by E . The increase in cost is $E \times C\underline{X}$ which must exceed the increase in the optimal cost $E \times (P[J] + Q - P[J+1])$. Division by E yields (3ii). If $X[J] > 0$, the same argument holds with $E < 0$. Therefore, $X[J] > 0$ implies $C\underline{X} - Q = P[J] - P[J+1]$.

It is useful to interpret $C\underline{X} - Q$ as an opportunity cost in a related optimization problem that obeys the flow conservation constraints (1) but ignores the total balance constraint (2). If (X,U,V) solve the original cost minimization problem subject to (1) and (2), then (X,U,V) will solve the related problem with opportunity cost $C\underline{X} - Q$ and no total balance constraint. In this way, Q acts as an incentive for holding cash. If the incentive is just right then the opportunity loss will be reduced so that the optimal program automatically satisfies the total balance restriction.

V. Examples

Several simulations are shown below. The data and length of the planning horizon were selected to illustrate the output and to ease interpretation of the results. In the output format, the first row numbers the time periods, the second row contains the cash inflows (S), the third the asset to cash transfers U, the fourth the cash to asset transfers V, the fifth the cash levels X, the sixth the level of the interest earning account, the seventh the costs incurred in each period, and the eighth and final row contains the dual variables P. The final column contains averages. To obtain the total cost, the average cost should be multiplied by the number of periods. The number in the last column, row eight is Q, the increase in optimal cost of increasing the *total* cash holdings by one unit. In addition, each of the three examples is preceded by a data statement that gives the relevant values of the global variables.

In each example the following global variables remain constant.

$$X\underline{Q} \leftrightarrow 5000 \qquad Y\underline{Q} \leftrightarrow 20000 \qquad C\underline{U} \leftrightarrow .03$$

$$C\underline{U} \leftrightarrow .04 \qquad \underline{R} \leftrightarrow .1 \qquad O\underline{E} \leftrightarrow 1$$

$$B\underline{P} \leftrightarrow 6 \qquad D\underline{P} \leftrightarrow 1$$

The exogenous cash flow S remains the same in each example.

$$S \leftrightarrow {}^{-}20000 \quad 8000 \quad 5000 \quad {}^{-}15000 \quad 10000 \quad 20000$$

In the first example, $L\underline{X} \leftrightarrow 0$, $A\underline{X} \leftrightarrow 5000$, and $C\underline{X} \leftrightarrow .1$. The result is shown in the tableau at the top of the next page. Note that $P[2] = .04$ and $P[4] = {}^{-}.03$. This implies total costs could be reduced by delaying one unit of input from period 2 to period 4. In contrast, costs are increased if inflow is delayed from period 1 to period 2. Note that $Q = .06 < .1 = C\underline{X}$. Raising the total balance constraint will increase the holding of cash in some period. However, the cost of this is less than the opportunity cost since the timing of additional holdings can be selected to save on transactions costs.

The second tableau sets $A\underline{X} \leftrightarrow 0$, and $L\underline{X} \leftrightarrow 5000$. The result average balance is 5000, however, costs increase due to a loss of flexibility. The optimal policy matches the inflow with the transfers into the asset account.

In the third example $L\underline{X} \leftrightarrow 0$, and $A\underline{X}$ was increased to 10000. The value of Q in the first tableau indicates that average costs should increase by $5000 \times .06 = 300$. The increase is actually 490. Note that the value of Q increases to $.08$. This measure is a loss in flexibility due to the increase in the total balance constraint.

With the $L\underline{X} \leftrightarrow 0$, and the compensating balance constraint $A\underline{X} \leftrightarrow 5000$, the cost $C\underline{X}$ was reduced to $.05$. This resulted in no policy change over Tableau 1 although Q dropped to $.02$.

```
      EPOPTDATA
   XQ          YQ                  LX         AX                CX          CU          CV           R
 5000.000  20000.000              0.000    5000.000            0.100       0.030       0.040       0.100

      PRINT EPOPT S
TIME PERIOD                1             2              3              4            5            6            AVG
NET CASH FLOW       ¯20000.00       8000.00        5000.00      ¯15000.00     10000.00     20000.00       1333.33
ASSETS TO CASH       15000.00          0.00           0.00        7500.00         0.00         0.00       3750.00
CASH TO ASSETS           0.00       5500.00           0.00           0.00     10000.00         0.00       2583.33
CASH STOCKS              0.00       2500.00        7500.00           0.00         0.00     20000.00       5000.00
ASSET STOCKS          7000.00      13200.00       14520.00        8472.00     19319.20     21251.12      13960.39
CASH COSTS             450.00        470.00         750.00         225.00       400.00      2000.00        715.83
MARGINAL COST           ¯0.03          0.04           0.00          ¯0.03         0.04         0.03          0.06
```

```
      EPOPTDATA
   XQ          YQ                 LX              CX              CU          CV          R
 5000.000  20000.00          5000.000            0.100           0.030       0.040       0.100

      PRINT EPOPT S
TIME PERIOD                1             2              3              4            5            6            AVG
NET CASH FLOW       ¯20000.00       8000.00        5000.00      ¯15000.00     10000.00     20000.00       1333.33
ASSETS TO CASH       20000.00          0.00           0.00       15000.00         0.00         0.00       5833.33
CASH TO ASSETS           0.00       8000.00        5000.00           0.00     10000.00     20000.00       7166.67
CASH STOCKS           5000.00       5000.00        5000.00        5000.00      5000.00      5000.00       5000.00
ASSET STOCKS          2000.00      10200.00       16220.00        2842.00     13126.20     34438.82      13137.84
CASH COSTS            1100.00        820.00         700.00         950.00       900.00      1300.00        961.67
MARGINAL COST           ¯0.03          0.04           0.04          ¯0.03         0.04         0.04          0.00
```

```
      EPOPTDATA
   XQ            YQ                LX         AX                CX          CU          CV           R
 5000.000     20000.00            0.000   10000.000            0.100       0.030       0.040       0.100

      PRINT EPOPT S
TIME PERIOD                1             2              3              4            5            6            AVG
NET CASH FLOW       ¯20000.00       8000.00        5000.00      ¯15000.00     10000.00     20000.00       1333.33
ASSETS TO CASH       15000.00          0.00           0.00        2000.00         0.00         0.00       2833.33
CASH TO ASSETS           0.00          0.00           0.00           0.00       500.00         0.00         83.33
CASH STOCKS              0.00       8000.00       13000.00           0.00      9500.00     29500.00      10000.00
ASSET STOCKS          7000.00       7700.00        8470.00        7317.00      8548.70      9403.57       8073.21
CASH COSTS             450.00        800.00        1300.00          60.00       970.00      2950.00       1088.33
MARGINAL COST           ¯0.03          0.01          ¯0.01          ¯0.03         0.04         0.02          0.08
```

VI. Summary

We have described a two asset cash flow simulation model. Given a sequence of future cash inflows, institutional restrictions, and a cash management objective the model projects future cash transfers, the costs of these transfers and future cash and income-producing asset stocks.

The model assumes a deterministic inflow of cash. In many cash management problems with planning horizons of one year or less and planning periods of one week or more, the random component of cash inflow is relatively small. Cash flows that occur in the relatively near future are predictable in magnitude. By concentrating on the nonstationary deterministic component of cash flows, the model schedules the size and timing of major cash transfers in order to meet institutional requirements and optimize cash management objectives.

The assumption of a single income producing asset is a useful first approximation to the multiple asset case. In particular, if various assets have larger yields for longer holding periods, then the cash transfers and asset levels calculated by the simulation program can be used as inputs to a multiple asset scheduling subproblem.

Several uses for the simulation package were outlined in Section II. There is another, perhaps more important, use of the model. The flexibility and interactive nature of the model gives the user a feel for the cash flow process and the interaction of different policy variables. The decision maker, by using such a simulation model, learns how to analyze cash measurement policies. In addition to the greater confidence which usually results from the use of such interactive models the user should also be able to detect the sensitivity of new policies to various institutional assumptions and management objectives. Hopefully this may lead to recommendations as to how the cash management process may be better organized and controlled.

Session 12: Gaming and Man-Machine Simulation
Chairman: Richard Levitan, IBM Corporation

This session features three papers in the field of computer based simulation in which humans make decisions in a simulated environment. All are concerned with the use of such models for teaching purposes; however, they provide an interesting spread in types of learning which is expected to develop.

Papers

"Progress Toward A Proposed Simulation Game Base for Curricula in Decision Sciences"
Geoffrey Churchill and Edwin Heard, Georgia State University

"Interactive Budgeting Models: A Simulation Tool for MIS Education"
Theodore J. Mock, University of California and
Miklos A. Vasarhelyi, Pontificia Universidade Catolica doRio de Janeiro

"The Traffic Police Management Training Game"
Gay Serway, Allen Kennedy and Gustave Rath, Northwestern University

Discussants

G. C. d'Ans, IBM Corporation
Richard Staelin, Carnegie-Mellon University
E. G. Rodgers, University of Toledo

PROGRESS TOWARD A PROPOSED SIMULATION GAME BASE FOR CURRICULA IN DECISION SCIENCES

EDWIN L. HEARD
GEOFFREY CHURCHILL
OPERATIONAL GAMING GROUP
DEPARTMENT OF QUANTITATIVE METHODS, GEORGIA STATE UNIVERSITY

ABSTRACT

Use of a special purpose business simulation game as a laboratory vehicle throughout a decision science curriculum is proposed as a pedagogically useful device for achieving curricular objectives. Development of such a game and its requisite characteristics are described. The multi-level nature of the game dictates that major subsystems must exist at different levels in order to incorporate dissimilar decision situations confronting players in various courses. Modular design is the means by which certain multi-level features are incorporated and permits one to "tailor-make" the game for a particular application. Economies, analogous to overlays in FORTRAN, are achieved since only those program segments or subsystems representing degrees of decision complexity actually present must be stored and executed in core.

Many curricula have been designed to imbue students with the Decision Science philosophy. (The reader is probably familiar with these attempts under such titles as scientific management, operations research, management science, or quantitative business analysis.) For the most part, such programs have been charged with severely limited success in that they have not met the major objective of all Decision Science curricula: to train effective situational problem solvers. Instead, new graduates of such programs tend to be sophisticated theoretical modelers whose value to business and industry is limited by their narrow technical viewpoint.[1] Their subsequent paths of development are highly individual, and depend not only on personal characteristics, but also on whether they find niches in research-oriented organizations.[2]

CURRICULAR VEHICLES

Traditional Vehicles

Traditional vehicles used in Decision Science curricula include mathematical exercises, "word problems", cases and project assignments. While each of these contributes, to some extent, to the accomplishment of the aforementioned objectives, they are all severely limited. Mathematical problems are useful only for teaching theoretical nuances and mathematical manipulation, and no situational elements are involved. Word problems include a few more situational elements, but do not adequately reflect the dynamic characteristics required for situational problem solving. In addition, word problems tend to be so brief that neither the choice of technique, nor the identification of relevant data provides an important challenge to the ingenuity of the textbook-wise student. Cases may reflect situational peculiarities much better than either of the foregoing types of problems. Unfortunately, cases are static in nature and, based on the experience of the authors, relevant cases appear to be in short supply. Project assignments can offer students practice in situational problem solving but, due to typical course time limitations, don't provide students an opportunity to fully examine the situational impact of their recommendations.

Traditional Vehicles

Consideration of the limitations of traditional vehicles might lead to the conclusion that practical experience is the only instructional vehicle capable of meeting all the objectives of a Decision Science curriculum. However, this is not the case, since real-world experience is limited by its inaccessibility to most students. Furthermore, most significant real situational problems are too complex for students' initial learning experiences. There is, however, one additional instructional vehicle available which, when appropriately used in conjunction with the traditional vehicles, may meet all of the objectives of a Decision Science curriculum. This vehicle is simulation gaming.

A Laboratory Vehicle

The authors contend that: (a) a

simulation game can be used throughout a Decision Science curriculum to integrate the various topical areas and to provide a laboratory situation in which students can obtain experience in all phases of Decision Science;[3] (b) a game to be used for this purpose must have some very special characteristics not included in existing widely used games; and (c) a game with the requisite features can be and is being developed by the Operational Gaming Group at Georgia State University.

The other reasons for using a game throughout a Decision Science curriculum involve the general features and inherent student appeal of such a game. First, simulation games are dynamic in nature. This feature allows students not only to make decisions at different points in time in a competitive simulated environment, but also requires them to observe and live with the results of those decisions. Also, the documentation accompanying such a game gives the student experience in analytical examination of written descriptions of situational problems, and in screening management reports for relevant data. Further, use of a game throughout a program provides continuity not possible with the use of any other vehicle. The fact that a simulation game is somewhat less complex than "real-world" situations eases the transition to "real-world" problem solving. Still, simulation games can, by design, be complex enough to demonstrate the systemic interactions between decisions in several functional areas of the enterprise.

An aspect not to be omitted is the impact on the organization offering the curriculum; typically a school or department within a university. While simulation gaming has compelling advantages as a teaching vehicle for Decision Science, it is undeniably expensive. Two highly significant expense elements are game development and user (instructor) training. By the choice of a single game package as the vehicle for an entire curriculum, it is expected that considerable economies may be realized on a per course basis as compared with a totally independent choice of vehicle.

Even if none of the attributes mentioned above was present, the quality of student appeal in simulation games would make their use worthwhile. In our experience, no other vehicle appears to be capable of generating student interest and motivation to the extent that simulation games do.

In order to meet the objectives of a Decision Science curriculum, it is necessary to develop in students an awareness of the need for, abilities of, and limitations of quantitative techniques. Simulation games appear to have a unique ability to create student demand for relevant quantitative techniques[4]. This is probably because the simulation game may require students to make decisions in an environment where obviously relevant data is present but where no explicit instructions are given for its use. Thus overloaded with information, students then become highly receptive to quantitative techniques which lend structure to this environment and which provide information that can be used in arriving at decisions.

REQUISITE GAME FEATURES

It was mentioned earlier that the use of a single game throughout the curriculum would lead to economy of vehicle development. If the desirability of using simulation games in a Decision Science curriculum is accepted, philosophical considerations also dictate the use of a single game. Student participation should be concentrated on acquiring experience in decision making and on using quantitative and behavioral techniques as inputs to the decision process, rather than on deciphering the documentation of several different games. The single game limitation imposes some rather stringent requirements on the design of such a game. These requirements concern the simulated environment, complexity, flexibility, adaptability, and documentation. The game under development at Georgia State University will incorporate the requisite features outlined below.

Simulated Environment

Since the game is to be used throughout a curriculum, it will be desirable to be able to focus on different subareas independently or simultaneously. A generalized business environment is probably the only one which will adequately cover the various areas where decision problems arise. The specifics of the environment are less important than the requirement that it must be possible to display the environment at a variety of levels of detail.

An appropriately designed business environment should provide potential decision making opportunities at a variety of organizational levels ranging from repetitive short run operational decisions at a lower middle management level to top management strategic decisions of great long range consequence. On another scale, the decisions should range from relatively mechanistic ones in which the outcome of a given action is highly predictable, through decisions in which uncertainty is a factor, up to those where the unknown future competitive actions of an opponent are critical.

Complexity

Several complexity considerations are important. First, the game must be simple enough that the student is not overwhelmed by its intricacies in his initial experience with it. At the same time, it must be complex enough that the problems and solutions are not obvious. Finally, its potential must be sufficiently rich to sustain student interest throughout a curriculum.

Flexibility

The use of the same game throughout the program and the need to focus on different subareas independently and simultaneously have some powerful implications for the structure of the computer program. During the early stages of the curriculum, it will be desirable for the student to be exposed to all subareas simultaneously, but in a very simplistic fashion. During the intermediate stages, it will be desirable to focus on highly sophisticated decisions in one or a few subareas while suppressing complexity in the remainder of the subareas. During the final stage of the curriculum, it will be necessary to pull out all the stops and allow the game to operate in its most complex form. The need for flexibility requires that the game be modular and multi-level. If, for example, the primary focus is on the production area, there must be a marketing module which operates in simplistic fashion. However, since in other cases the desire may be to focus on marketing, a sophisticated marketing module must also be built. Still another flexibility requirement dictates that it must be possible to configure the game in such a way that outside "role play" can be superimposed on the operation of the game in a realistic and relevant manner. Likewise, it must be possible to incorporate the impact of one time outside effects such as strikes, anti-trust action and surtaxes for excessive retention of earnings.

Adaptability

Since the game must be operated in several different configurations, it is necessary that the changeover time and effort required be small. The game administrator should be able to choose the desired configuration easily; for example, by specifying one parameter value for each subarea, thus specifying a set of decisions which are to be held open or closed. Another arrangement might be through merging desired modules with a base program to create a custom-tailored program with the desired combination of simplicity and sophistication. While this would be rather more demanding of the instructor (who may not be assumed to be sophisticated in the technical aspects of gaming), a richer variety of options can be provided in this manner. In such fashion, the game can be tailor-made to each specific course in the curriculum.

Documentation

Naturally, operating the game in different configurations will prohibit the use of a single documentary unit for all applications. Consequently, documentary adaptability must exist to the same degree as does computer program adaptability. A background environment description should be written which is general enough to provide the student with a broad understanding of the general business environment of the game without detailing the specific decisions to be made. Then, for each module, a set of documentation must be prepared, one for each different level at which the module can be operated. These can also be coded so that the modules chosen by the game administrator to specify a particular configuration will also specify the appropriate background documentation and decision forms to be provided for the students. Additionally, an instructor's manual must be prepared to aid the instructor in choosing the game configuration which will best meet his objectives.

Game Development

Unfortunately, any single game in current use is woefully inadequate for use in such diverse configurations, although there are several games available which are admirably adequate for achieving specific goals within particular environments. This is not surprising, since most games were written to achieve selected objectives within given environments. Another limitation of existing games is the quality of the accompanying documentation. Many games have been written by individuals or by small teams. Such games and the accompanying instructions tend to reflect the special areas of interest of the authors. Coverage of areas peripheral to the areas of interest of the authors is given only superficial treatment . . . A final limitation of existing games is the level of sophistication required of the participants. Here again, special purpose games are often written for audiences at specific educational levels. As a result of the limitations of existing games, it is necessary for game administrators to learn, implement and administer different games in order to achieve specific purposes for differing audiences.[5]

Recognition of the limitations of

existing games for accomplishing the objectives of a Decision Science curriculum spurred the Operational Gaming Group within the School of Business at Georgia State University to embark on the development of a game which would include all of the requisite features above. The Operational Gaming Group consists of a number of faculty members, each of whom has expertise in some area important to gaming. Participants in the Group are: Geoffrey Churchill, Chairman; Sandra Beldt; Merwyn Elliott; David Ewert; Dennis Grawoig; Elbert Greynolds; Edwin Heard; Don Jewell; Arthur Nichols; Brian Schott; Dwight Tabor; and Jerry Wheat. The Group, centered in the Department of Quantitative Methods, is both interdepartmental and interdisciplinary with representatives from accounting, economics, finance, insurance, marketing, personnel management, and production management.

The Group decided to program the game in BASIC so that modules could easily be merged with the executive routine, and because of the time-sharing capability and widespread availability of the language. The Group then made two crucial decisions; the first version of the game was to include only relatively simple modules for each subarea; and development of more sophisticated modules would be accomplished simultaneously with the programming, debugging, and parameterization of the simplistic version of the game.

Various tasks were allocated to each member of the Group. The development of a rough flowchart of the executive routine was assigned to the Group member with the most gaming experience. Preparation of rough flowcharts for the individual modules was assigned on a one-for-one basis to Group members with expertise in the specific areas. All rough flowcharts were funnelled to one Group member for refinement and logic review while another Group member translated all flowcharts to BASIC and stored the program on the computer. Debugging was handled by a team of three, composed of the Group coordinator, flowcharter and programmer. Parameterization was handled by the group economist. Testing is taking place in two phases as additional features are added. First, the members of the Group play the game and suggest possible modifications. Second, the game is tested in a graduate class at Georgia State University.

At this writing (October, 1972), a great deal of work remains to be done to develop a game capable of supporting a curriculum such as that discussed below. Nevertheless, the authors feel that a substantial beginning has been made. The executive routine, which embodies the basic environmental model, is running reliably on the GSU UNIVAC 7 as are the initialization and output programs. These functions have been deliberately separated, due to the relatively small size limit frequently imposed on BASIC programs, in order that space be available for merging a number of modules simultaneously. (See Charts I and II.)

These routines incorporate a market of a fairly high order of complexity (as compared with existing marketing games),[6] a production process of quite moderate complexity[7], financial decisions of a rudimentary sort[8], and fairly detailed accounting reports[9]. Additionally, some modules have been developed and tested. These include modules for Research and Development, Marketing Research, Personnel Evaluation and Fixed Asset Acquisition (equipment replacement/plant expansion). A financial accounting module which will permit examining the effects of a vast variety of reporting procedures on accounting information is nearly ready. Work has begun on a module to expand the scope of financial decision-making. Personnel have been assigned to modules at higher complexity levels in both marketing and production.

In a related project, the existing programs have been class tested by Geoffrey Churchill and Sandra Beldt, with a view to implementation of the game in what will be described below as the "Early" core. (Note that this does not presuppose implementation of an entire game-based curriculum. This course is presently based on an excellent game,[10] but one designed for MBA use.) Despite break-in problems of a normal sort, it is fair to comment that initial expectations appear to be met; students did grasp for models as a satisfying way of bringing order out of chaos.

PROPOSED CURRICULUM

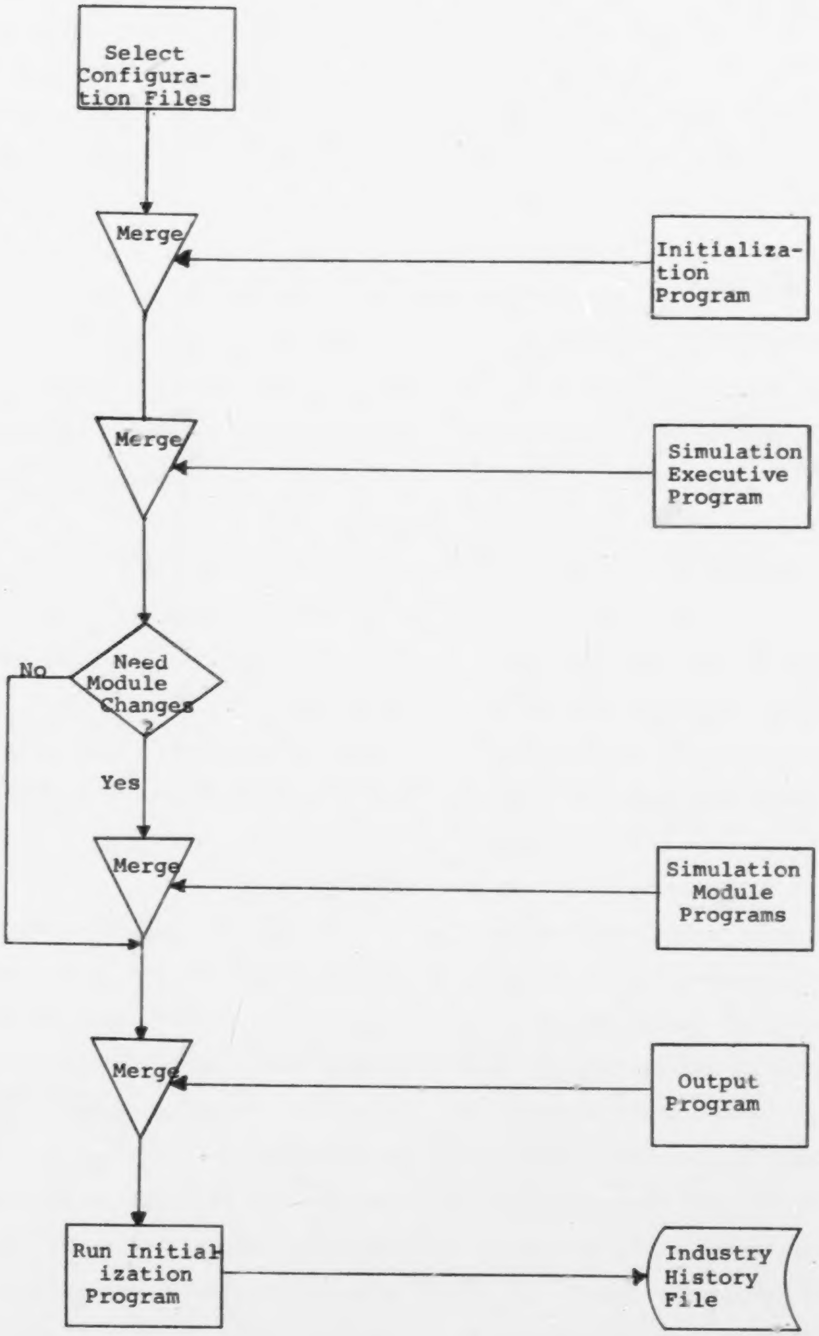
The proposed curriculum consists of three major components: an "Early" Core, electives, and a "Late" core. Chart III illustrates the precedence relationships between the three components of the program.

"Early" Core

In the "Early" core, the student is introduced to the game, remote terminal time-sharing, systems concepts, basic modelling in a dynamic environment, and canned programs. After a preliminary introduction to the game, the student is required to play a moderately complex version of the game with a relatively fast rate of decision making. The anticipated result is im-

CHART I

GAME PREPARATION FOR COURSE USE

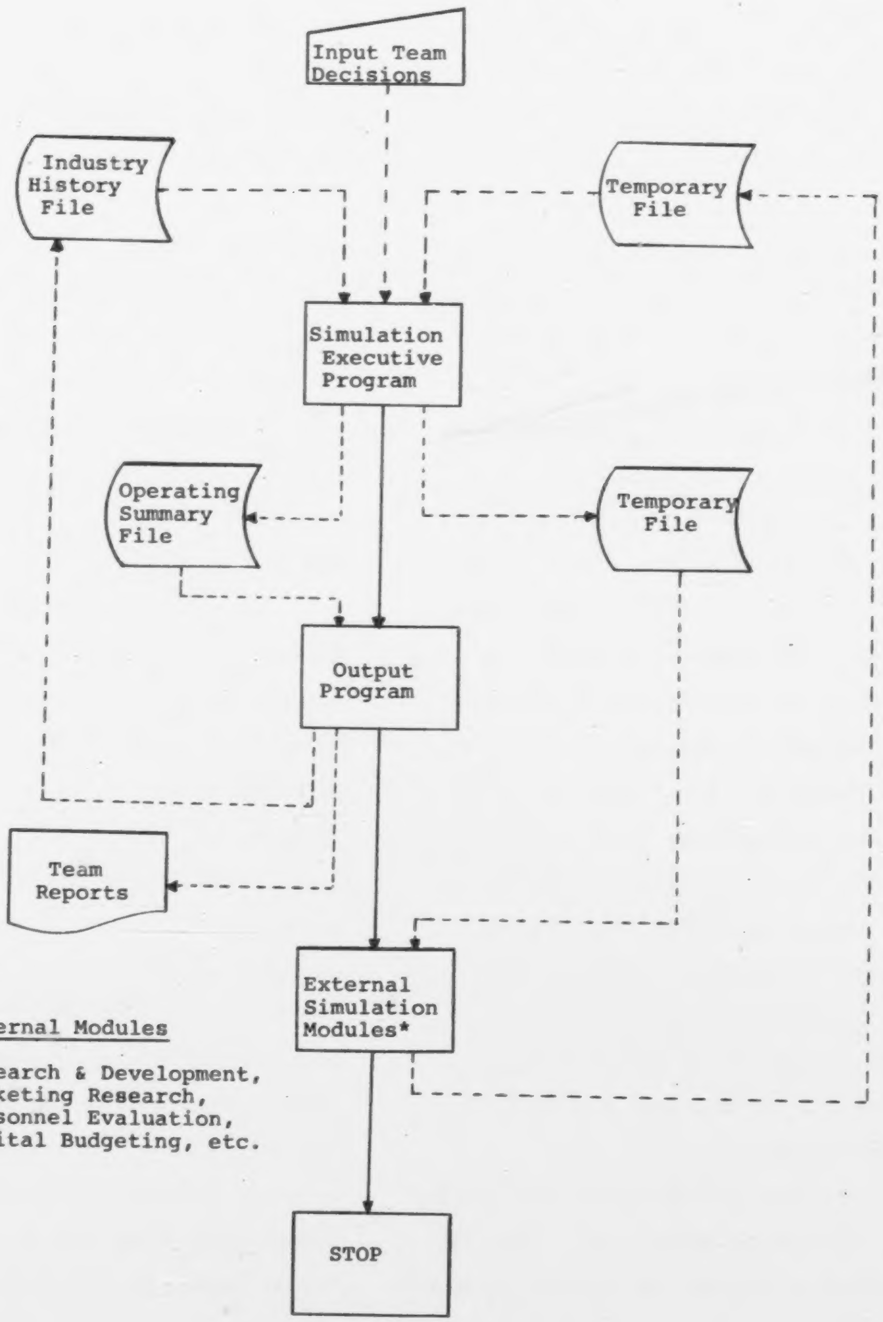


Select Configuration Files
Merge
Initialization Program
Merge
Simulation Executive Program
Need Module Changes ?
No
Yes
Merge
Simulation Module Programs
Merge
Output Program
Run Initialization Program
Industry History File

CHART II

SINGLE DECISION PERIOD SIMULATION

GAME OPERATION


Input Team Decisions
Industry History File
Temporary File
Simulation Executive Program
Operating Summary File
Temporary File
Output Program
Team Reports
External Simulation Modules*
STOP

* External Modules

Research & Development,
Marketing Research,
Personnel Evaluation,
Capital Budgeting, etc.

mediate chaos. The student is next introduced to some basic quantitative models, which will hopefully bring order out of the chaos.

These basic models include forecasting, cash flow, regression, inventory and linear programming. Each model is introduced in a survey fashion; the canned programs available for them are presented and ways in which they can be used to provide information for decision-making are presented. Students are expected to adapt these models to their needs in the game context, thus getting practice in all phases of Decision Science: observation, problem identification, description of relevant relationships, experimental investigation, interpretation of experimental results, and translation of information into effective action[11].

Before a student uses any model for decision-making, he is required to present a proposal that the model be built to a management committee consisting of students from the "Late" core for their criticism and evaluation. After he has his proposal approved, he is required to file an implementation report.

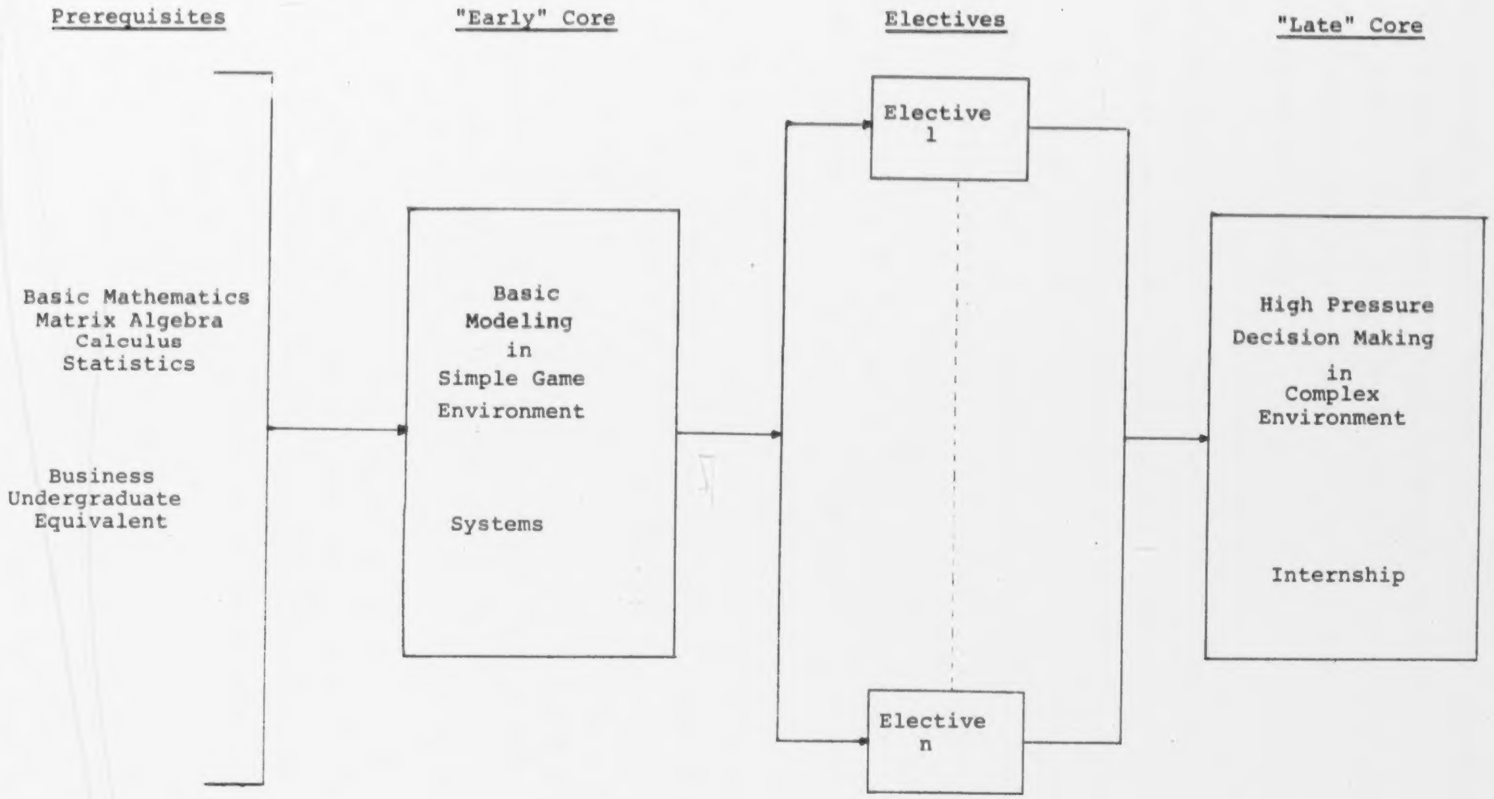
The systems courses in the "Early" core have three objectives. They are expected to enable the student to develop a view of the organization as a system and to recognize and understand its information supplying and suppressing characteristics. A simpler version of the game is used which reports only financial statistics and allows for the expansion of a product line. Students are required to design their own information systems for decision making purposes and to trace the systemic implications of product line expansion from the opening of a research and development facility to product maturity. "Late" core students are again used to criticize information systems designs and suggest revisions.

Elective Courses

These courses are more nearly comparable with currently existing Decision Science courses than are either the "Early" or "Late" core. Still, there are substantial differences. The objectives of these courses include expanding and intensifying the students' modelling capability; indoctrinating the students with a feel for data problems; developing student understanding of the non-independence of quantitative techniques; acquainting the students with the more subtle capabilities and limitations of models; improving the students' understanding and appreciation of sensitivity analysis; and bridging the gap

CHART III

DECISION SCIENCE CURRICULUM


Prerequisites
"Early" Core
Electives
"Late" Core
Basic Mathematics
Matrix Algebra
Calculus
Statistics
Business
Undergraduate
Equivalent
Basic
Modeling
in
Simple Game
Environment
Systems
Elective
1
Elective
n
High Pressure
Decision Making
in
Complex
Environment
Internship

between model solution and decision making.

Course construction to accomplish these objectives is difficult but possible. Lectures can be developed which include a solid theoretical approach through logic and pictures, while escaping the mathematical derivation-theorem proving time sink. Situational problems can be utilized which require ingenious parameter development and allow for prediction of results of decisions. Students can be introduced to canned programs which can be used via remote terminal time-sharing to facilitate any required computation. Available practical computational algorithms should be reviewed along with sources and special features of different packages.

Here, too, the game has a very definite role. It can be used in the appropriate configuration for each course to demonstrate and provide specific course related situations where particular techniques are applicable. In general, however, the game would be incidental to, rather than dominant in, elective courses.

At this point, perhaps some examples are in order. A game configuration which required product mix, transfer pricing, transportation, warehousing, and aggregate scheduling decisions would provide an excellent environment for the introduction of mathematical programming. A configuration which required inventory, cash flow and product line expansion decisions would provide an equally good environment for the introduction of simulation. And what better environment in which to introduce multivariate statistics than one which required pricing decisions and forecasts?

"Late" Core

These courses are designed to provide the transition from student to decision scientist. The first of the "Late" core courses would involve full-scale use of the game with every compatible option present. Students would be required to perform and report to a management committee on major studies of topics such as the automation of routine decisions and the upgrading of return on investment. These groups would also be required to investigate the feasibility of, institute where indicated, and review plant expansion decisions, new product introductions, competitor acquisition decisions, and other decisions of major strategic impact.

The final course in the curriculum would require each student to serve an internship with a local business. During this internship, he would be required to choose a "real" problem, use the decision science approach to solving the problem and present his project in management recommendation form to a committee composed of members of the Decision Science faculty and a representative of the firm where he interns. The student's only other responsibility during the period would be to serve on management committees for "Early" core courses. This feature of the curriculum would complete the student's transition from student to decision scientist and would provide him with a brief but tantalizing view of the other end of the decision making spectrum.

CONCLUSIONS

Progress in the direction the authors have suggested here is highly desirable, but should not be rushed. It was probably the undue haste of various schools in entering the quantitative decision making arena that resulted in the "sophisticated theoretical modeler" phenomenon alluded to in the first paragraph. The game with all the requisite characteristics outlined herein is essential to the success of a curriculum such as the one proposed in this paper.

To the authors' knowledge, an appropriate game does not currently exist[12]. The Operational Gaming Group at Georgia State University has one partially completed. If an existing game with the requisite characteristics has been overlooked, the authors would appreciate any available information on it.

ENDNOTES

[1] Halbrecht Associates, Inc., "Industry's Changing Personnel Specifications for Operations Research Positions," Washington, n.d.

[2] Naturally, there are exceptions to this generalization, such as those people who successfully move into line management.

[3] This claim presupposes the availability of time-sharing remote terminals and canned programs for student use. The contention is based on the use of the game as a learning vehicle rather than as a teaching medium.

[4] See for example, Brian Schott and Arthur C. Nichols, "The Use of a Business Simulation Game in an MBA Core Course in Quantitative Methods," in Laboratories for Training and Development of Executives and Administrators, Oklahoma Christian College, Oklahoma City, 1971.

[5] Edwin L. Heard and Geoffrey Churchill, "Operational Gaming Group Formed at Georgia State," Simulation/Gaming/News, Vol. 1, No. 2, May, 1972.

[6] For comparison, see Ralph L. Day,

Marketing in Action (rev. ed., Homewood, Illinois: Richard D. Irwin, 1968).

[7] For comparison, see Geoffrey Churchill, *JØBLØT* (New York: Macmillan, 1970.

[8] For comparison, see Paul S. Greenlaw and M. William Frey, *FINANSIM* (Scranton, Pennsylvania: International Textbook Company, 1967).

[9] For comparison, see Bill R. Darden and William H. Lucas, *The Decision Making Game* (New York: Appleton-Century-Crofts, 1969).

[10] Arthur C. Nichols and Brian Schott, *SIM^{Q}, A Business Simulation Game for Decision Science Students* (Dubuque, Iowa: Kendall/Hunt, 1972).

[11] See Ronald A. Howard, "Management Science Education: Nature and Nurture", in *Management Science*, Vol. 17, No. 2, October, 1970. Howard suggests that the course program should include as key elements of study uncertainty, complexity, dynamic effects, economics, optimization, modeling and computer analysis, and behavioral science.

[12] Thanks to an un-named reviewer, we have discovered that groups at Wake Forest and Carnegie Mellon are developing business games which allow for the sequential introduction of decision areas. We understand that these games would, in effect, provide two levels of complexity for each decision area, i.e. each decision area is either there or it is not.

INTERACTIVE BUDGETING MODELS:
A SIMULATION TOOL FOR MIS EDUCATION

Theodore J. Mock

University of California, Los Angeles

and

Miklos A. Vasarhelyi

Pontificia Universidade Catolica do Rio de Janeiro

Abstract

A relatively new and innovative educational approach in graduate information systems is discussed. The thrust of the approach is to have student groups design computer-based simulations of the budgeting processes of a firm in the form of an interactive budgeting model. These are applied to the areas of financial planning, control and managerial decision making. Several different approaches were suggested to and adopted by students as main design philosophies including modular planning and a matrix accounting system. The various implemented systems are described and their features are classified.

In its attempts to keep up with practice, the academic world is ever striving to develop improved pedagogical techniques and thus better-prepared information systems students. This paper reports on such a device, an Interactive Budgeting Model (IBM), used in the accounting-information systems program within the Graduate School of Management.

One goal of this program is to expose students to online planning systems as described by Sackman and Citrenbaum[1]. If more and more implemented information systems incorporate online planning and control features[2], pedagogical adaptation is necessary. In this situation, an educational problem is to expose accounting and MIS students to the methodologies, man-machine interface, complexities and problems of system design in an academic environment. The approach described here is based upon student planning and implementation of a simulated, financially-oriented information system. The financial, budgetary focus of the project narrows the scope of possible projects and also has an additional benefit of requiring an in-depth knowledge of the accounting process (a skill frequently lacking in MIS and MBA students). A matrix accounting model is proposed to form the backbone of the system.

---

[1] Sackman, H. and R. L. Citrenbaum (Eds.), Online Planning, Prentice-Hall, 1972.

This paper is partitioned into three main sections including this introduction. The following section discusses the concept and implementation features of an Interactive Budgeting Model in terms of its educational objectives, its simulation features and its modeling features. The third main section of this paper describes actual projects designed and implemented by students at UCLA and Ohio State University in three different courses. This summarization is intended to give the reader an idea of the specific features designed into these systems and their shortcomings. In addition, a time series comparison is presented of the features used at UCLA and Ohio State University.

Interactive Budgeting Models

The implementation of an Interactive Budgeting Model is the focus of approximately fifty percent of the student's effort in a graduate class at the masters level at UCLA. This class, called Information Systems for Planning and Control, is oriented towards exposing students to the problems of information systems design and corporate planning and control. This course is the second in a series of three where the first is oriented towards systems theory and the systems approach and the third is oriented towards problems of measurement in information systems.

In response to inadequacies in case, discussion and lecture approaches to the course, a tool was sought that would provide the experience students needed for systems analysis and system implementation of corporate financial information systems. Such a tool should have experiential features in which the student would encounter the problems inherent in systems design and analysis and also which would exhibit planning and control concepts. Such specifications led to the concept of the IBM, an interactive system for planning and control in a simulated environment.

---

[2] For example, Redwood, P.H.S., "APL For Business Applications," Datamation, May 1972, pp. 82-84; and "Litton's Electronic Information Machine," Business Week, March 28, 1970, pp. 158-160.

Students were instructed to design and implement a conversational system which would allow a manager to interact with a terminal and assist in management decisions. The nature of this task and the boundaries of the problem were purposefully left ill-defined as the problem specification and contextual design phases are important experiences in the desired educational process. For example, Pounds[3] points out that "seldom if ever, do managers analyze or understand the sources of their problems," and "...the availability of formal problem solving procedures serves only to highlight those parts of the manager's job which these procedures do not deal: problem identification, the assignment of problem priority and the allocation of scarse resources to problems." Therefore, in an education process, accurate definition of a problem may hinder education in an area where the manager often is lacking.

Beginning with an ill-defined problem specification, students were then given several basic references[4] that may be useful in constructing their IBMs and were told that some quantitative techniques such as PERT, linear programming or regression may be profitably incorporated in the model.

The first two or three weeks of the class were then dedicated to teaching an appropriate interactive computer language. At UCLA, APL was used. This powerful interactive language is easy to learn and students, working in groups, tend to make up for their individual deficiencies.

Interactive computing and debugging permits students to obtain fast and accurate performance feedback on the main features of their models. In designing their system, the following were suggested as minimal design criteria:

1. Security features
2. File management of historical accounting data
3. A useful interactive decision aid
4. Planning and control features
5. Modular approach, matrix accounting structure.

Also, a list of possible modules for the IBMs were given to the students:

---

[3] Pounds, W. F., "The Process of Problem Finding," The Industrial Management Review, Vol. II, No. 1, Fall 1969, pp. 1-21.

[4] Mattessich, R., Accounting and Analytical Methods, Richard D. Irwin, Homewood, Illinois, 1964; Butterworth, J. W. "The Accounting System as an Information Function," Mimeograph, University of British Columbia, June 1970; Ness, D. N., "Interactive Budgeting Models: An Example," Working Paper, M.I.T.

1. Output module
    - Display of financial statements, projected budgets, network schedules, selected display of underlying planning assumptions (e.g., rate of growth)
2. Input module
    - Reading and storing information
    - Building data bases
    - Retrieving information
3. Performance analysis modules
    - Calculating performance analysis ratios
    - Preparing output and management exception reporting
4. Incorporating transactions
    - Measurement of economic events
    - Periodic reporting
5. Building and using management science functions
    - L.P., statistical analysis, graphics, charts, discounting
6. Control and security features
7. Specific planning and forecasting aids
    - Regression, exponential smoothing, consensus (Delphi) techniques
8. Programmed decision rules
    - Exception limits, cash constraints

As is demonstrated in a following section of this paper, these suggested module specifications inspired a large variety of IBMs.

In implementing the tasks involved in constructing such an IBM, the student groups had to consider a series of system design problems including problem and project specification and management. These are discussed from a pedagogical viewpoint.

First, the student team had to decide on what type of organization and which specific planning and control problems they wanted to model. Some chose a simplified model of an entire firm or department and concentrated on implementing several management science techniques. Others decided to simulate a small part of a large system and attempted to attack its problems extensively. Part of the learning gained from this step in the model development is the need to specify, limit and dissect the possible problems to be tackled. A main cause of difficulty and frustration in such projects was tackling too large a problem and the eventual difficulty of implementation within time constraints.

MIS classes usually draw students with a variety of backgrounds including accounting, computer methods, marketing and behavioral science. Such heterogeneity results in a labor distribution problem. Frequently, students specialize such that those interested in computer methods concentrate on programming while accounting majors study information flows and reporting techniques. In contrast some groups divide programming tasks evenly among their constituents. Either approach frequently generates serious coordination problems as certain parts of the project are completed on schedule and others are not.

Once a group settled on a problem area, problem focus was needed. Groups frequently tried to overachieve and during the later stages of the project began to realize that their objectives were not realistic and should be redefined. Part of the difficulty of the instructor's task was to warn students about such risks without undercutting the learning potentiality of these experiences.

The experience that an IBM project lends to the students in terms of group processes is certainly an important educational aspect in MIS education. Although there was a definite task and deadlines and assignments in the beginning of the course, there was often too many ideas and little consistency among them. Also there were emerging leadership patterns and conflict for leadership roles. Evidence of this was that groups sometimes could not reach consensus and attrition occurred. Management and coordination problems always seemed to occur. Students frequently experienced the point that Argyris[5] makes "... the introduction of a sophisticated information technology is as much an emotional human problem that requires interpersonal competence (as well as technical competence) and that requires knowledge about the human aspects of organizations such as personality, small groups, intergroups and living systems of organizations norms."

The Underlying Matrix Accounting Structure

The IBM concept has been suggested[6] as a technique to design surrogate information systems and decrease software development costs. In this era of rapid change in which education has lagged technological development, new tools for education are needed. Many principles in the design of large scale software information systems are not theoretically sophisticated and may even be counterintuitive in nature due to the intricate interconnection of different system components.[7] The same may also be true in the design and integration of large scale systems where a large number of components interact and factors are interconnected. In such a situation the utilization of simulation technology for education and for the design of large scale software systems seems to exhibit great value.

---

[5] Argyris, Chris, "Management Information Systems: The Challenge to Rationality and Emotionality," Management Science, Vol. 17, No. 6, February 1971, pp. B-275-292.

[6] Vasarhelyi, M. A., "Simulation - A Tool for Pre-Implementation Testing of Large Scale Software Systems," Winter Simulation Conference, 1971.

[7] Forrester, Jay W., "Alternatives to Catastrophe - Understanding the Counter-intuitive Behavior of Social Systems," Technology Review, January 1971.

The matrix approach to accounting is suggested as the backbone of the IBM for several reasons, some of which have been alluded to earlier. Essentially a matrix model reveals the entire (wholistic) impact of each actual (or planned) accounting transaction on the entire set of financial reports or budgets. Thus the student must relate to the entire financial system and the impact of planning assumptions and decisions upon this system.

The matrix accounting approach considers entries in the firm's chart of accounts as a vector of period transaction amounts (T) multiplying an incidence matrix (I) composed of zeros and plus or minus ones.[8] The vector is equivalent to the possible accounting actions in the firm and the incidence matrix indicates which of the accounts of the firm are affected by such financial transactions. All account balances (B) at the end of period t are given by the identity $B_t=T_tI+B_{t-1}C$, where C is a matrix which closes nominal accounts. For a computerized system t may represent "real time," a day, a week or whatever and T may be projected (for planning) or actual financial transactions. The matrix approach can be extended to include policy changes in the firm. Such a methodology allows users to design and, through a simulation, consider the effects of policy changes over projected, proforma or budgeted financial statements. Interesting effects can be obtained by augmenting such models with OR techniques and interrelating interactive policy changes with partial optimization of system parameters. The utilization of simple linear projections can be made more realistic by the utilization of exponential, logarithmic or exponentially smoothed functions. Clearly, however, the monitoring features of interactive simulation are advantageous as the type of projections can be adapted to the realistic overview of the manager. Such powerful tools have been utilized in the construction of IBMs as are described in the next section of this paper.

Implemented Models

At this time, ten student IBMs have been designed. In the remaining part of this paper, these systems are summarized and contrasted. Focus is placed upon the underlying computer language and system characteristics and upon implemented simulations and man-machine considerations.

The projects will be discussed in terms of the three different classes where this technique was utilized. Computer capabilities and environmental factors were somewhat different for each class, thus providing an interesting comparative, longitudinal study. The first projects were developed during the spring quarter of 1971 at UCLA only four months after APL had been "brought up" on campus. The IBMs were implemented on the university's IBM 360/91 using APL which, at that time, had neither file nor fast formatting capabilities. As APL was the only interactive language available at UCLA, language choice was not a problem.

---

[8] Butterworth, J. W., op. cit.

The second set of projects were developed at Ohio State during winter quarter, 1972. At that time, computing was carried out on an IBM 370/165 and the Time Sharing Option (TSO) was up and running. Although BASIC and TSO FORTRAN and PL/1 were available, all student groups selected the CPS (Conversational Programming System) language which is essentially an interactive subset of PL/1. In comparison to UCLA, the main constraints of the OSU system capabilities were: 1) limited selection of business oriented preprogrammed subroutines, 2) in comparison to APL, CPS is a less powerful language and requires considerably more code, and 3) relatively slow response times. On the positive side, the OSU system was more stable and provided file capabilities so necessary for any realistic financial data base.

The third set of projects were developed during spring quarter 1972 at UCLA. This group used the same computing hardware previously described except by this time a preprogrammed fast formatting routine and file capabilities were available.

In terms of overall results, the scope, insight and technical quality of the IBMs was quite impressive, even for groups of graduate MIS students. This is evident in the sample material that follows. As expected, most problems were related to intragroup conflict or inability to establish realistic project goals and scope. The rather tight schedule facing the students was as follows:

Course Week	Topic and Assignment
1	General, but purposely vague, description of IBM concept, project requirements, and possible design criteria
1,2, & 3	Review (learn in many cases) appropriate computer language
3	Project plan and description, including PERT-type schedule, due and presented during class (this of course facilitated cross fertilization of ideas)
6	Oral report on progress and problems
9	Class demonstration of completed IBMs
10	Written reports including sample output, documentation and project critique

Project Summaries

As one would expect, a wealth of data exists on the ten projects. In an attempt to reduce these data, a taxonomy of each IBM is included in Table I. The taxonomy includes available computer system capabilities, description of the simulated entity, IBM modules, planning and control features, system features and problems.

Upon examining this summary, the following patterns emerged. First, the latter IBMs are more sophisticated than the earlier ones. Probably this was due to an increasing ability on the instructor's part to describe alternatives. Another factor was the improved system capabilities, particularly file management.

Although simulation modules were suggested either to incorporate environmental uncertainty or to test strategy alternatives, no group implemented such a module due to its intrinsic complexity. More importantly, design of the man-machine interface was neglected. For example, the conversational features of the executive modules were generally inadequate. This is disconcerting as a suggested systems criterion was "a useful interface decision aid," i.e., user oriented. One explanation is that in the initial stages of such projects, participants tend to be "systems biased" and thus they focus on implementing the forecasting, accounting, data base and reporting systems. Such oversights and biases were fed back to the groups as part of the learning process.

Sample Interactive Budgeting Models

A sample of some of the IBM materials follow. These include a project plan (Exhibit 1), a CPM-type schedule for an IBM (Exhibit 2) and sample output of two systems.

TABLE I

TAXONOMY OF PROJECT CHARACTERISTICS

Computer System and Options	Organization Description: Simulation of Real or Fictitious Firm	System Modules (*indicates planned but not implemented)	Other Planning and Control Features	Design Features and Problems, Miscellaneous Data
Class 1, UCLA, Spring 1971				
IBM 360/91 APL without fast formatting options 32K workspaces	Group 1 Ficticious TV manufacturer with 3 production lines (Risk Inc.)	Operational Control Optimization of production re L.P. Effect on budgets, sales forecasts Management Control (MC) Retrieval of current financial statements Stochastic sales forecast Ratio analysis of financial budgets Strategic Planning (SP) Present value analysys of investments GNP based long run sales forecast	User can resort to and test a variety of planning assumptions. Overall function TOT integrates other modules and facilitates comparison of forecasts derived from SP and MC including implications.	System Requirements Needed more than 1 workspace, WS FULL ERRORS. Use of preprogrammed LP and REG library routines. Developed from total systems philosophy. What-if (sensitivity) analysis implemented.
	Group 2 Ficticious, Small retailer	Sales forecasts and derived financial budgets Financial reporting Control (planned only)	Planning: Complete transaction based financial plans (flexible horizon)	1 workspace No file or security considerations Output fairly well formatted Control features (module) incomplete Limited decision aid Sensitivity aid
	Group 3 Ficticious, Plate Glass Co., focus on 5 process production dept.	Forecasting Sales, Cash Flow (4 periods) I/S and Balance Sheet Sales forecast based on macro economic data, historical company data, e.g. UCLA business forecast		1 workspace Poorly formatted output Internally (WS) managed data file

TABLE I (CONTINUED)

Computer System and Options	Organization Description: Simulation of Real or Ficticious Firm	System Modules (*indicates planned but not implemented)	Other Planning and Control Features	Design Features and Problems, Miscellaneous Data
	Group 3 (Continued)	Process cost analysis (including change of costing rates) (cost and transfer price) Variance analysis of budget to historical and actual to budget (both absolute and relative)		
		Class 2, OSU, Winter 1972		
IBM 370/165 CPS language (Conversational Processing System, subset of PL/1) Operated under TSO Some File Capabilities, Limited Preprogrammed Public Routines	Group 1 Real Highway Construction Company focus on contract bidding- cost control and performance analysis. Simplified contract developed.	*L.P. generation of optimal bid Base generation and management Data base retrieval of cost and interactive bid generation Cost measurement, variance reporting and data base update		Significant code was required, especially for file management. Integration was not accomplished. Significant file loss problems were encountered. Response time was a problem, particularly to retrieve file data.
	Group 2 A-Loss Co. Ficticious, sing single product manufacturing company.	Sales forecast Financial forecast (overall budget) Transactions (accounting) Control and performance reports Executive (overall integration and sensitivity analysis and security)	Sensitivity of budget-to-budget assumptions (e.g. collections schedule) Management by exception	System derived constraints: Quarterly updating Maximum of 30 accounts and 20 transactions Maximum of 5 systems functions Other desirable features: Dating function Audit trail Security log.
	Group 3 MBA Co., Real Ohio non-ferrous jobbing foundry	Data base (actual historical data) *Integrated cost control system Accounting module Regression analysis and forecasting	Monte Carlo Simulation of financial results with interval estimates.	Information needs derived from management needs. Design focus on what-if capabilities. Integration of modules never really completed.

TABLE I (CONTINUED)

Computer System and Options	Organization Description: Simulation of Real or Fictitious Firm	System Modules (*indicates planned but not implemented)	Other Planning and Control Features	Design Features and Problems, Miscellaneous Data
	Group 3 (Continued)	Budgeting Interactive Control (executive) Report generator		Significant file errors and loss.
		Class 3, UCLA, Spring 1972		
IBM 360/91 APL/360 with PLUS/ FILE option and fast formatting (ΔFMT) 48K workspaces	Group 1 Real Organization AISRP Research Center, UCLA	Security Budgeting Control	Comparative budgets (updated vs original)	Simplified funds-oriented accounting structure
	Group 2 Real firm construction contracting, fictitional data	Executive Planning Transactions Control	Regression forecast of So. California construction. Cash focused financial control.	Designed re interview of management's information needs. Desirable additions include user-control over forecast parameters, explicit sensitivity comparisons.
	Group 3 Real problem, planning and control system for movie production	File creation (selection of movie activities and levels) File Update (actual data) Cine-budget Printout	PERT/Cost schedule Progress and variance reports Completion performance reports	Extensive file utilization.
	Group 4 Real organization, Health research project management	Security Executive query Data base management Management module	1-year program budget, 5-year program forecasts Minimal control implemented	Program budget (PPB) orientation Focus on information retrieval rather than decision aid Significant module interface problems System output bound.

EXHIBIT 1

Project Plan, Group 2, Spring 1972, UCLA

OBJECTIVE

The objective of our project is to design an interactive budgeting and control system to meet the information needs of the President of an insulation contracting firm.

SCOPE OF OUR DESIGN

The system that we are designing is concerned mainly in providing the President with information to help him in the area of management planning and control. It designed around an analysis of his major decisions in that area, the process he uses to make those decisions, and the information he feels he requires. Since the manner in which each branch is handled is similar, we have designed the system with reference to only one branch with the assumption that by duplication it could easily be expanded to handle all the branches. We have broken the system into five major modules which are described below.

MAJOR MODULES AND THEIR FUNCTIONS

Forecast Module

The forecast module is to be used to estimate quarterly sales for the next 12 months. This will then be broken down into monthly sales. The method of linear regression is to be applied, utilizing those factors that we feel are appropriate - interest rates, building permits,etc. Sensitivity analysis will be available to the manager. Using sales forecast as a basis we will then be able to forecast other accounts and prepare budgets.

Transactions Module

The Butterworth Matrix System of accounting will be used in this module to handle the traditional accounting transactions and produce the monthly and yearly financial reports.

Control Module

This module will produce the needed reports to show variances from the budgets or standards. Certain general reports will be produced automatically at the end of each month or whenever desired, and other more detailed reports will be available on request.

General Reports to Be Produced

Variation in Income statement and Balance Sheet accounts as
   compared to budget - monthly and year to date.
Cash forecast for next three months
Signed sales contracts for next three months as compared to
   estimated sales for next three months.
Profit as compared to budget and previous year.
ROI as compared to expected standard
Changes in current ratio
Changes in working capital
Accounts Receivable and Inventory turnover as compared to standard
Inventory level as compared to standard

Security Module

This module will handle security procedures to insure that it cannot be accessed by unauthorized individuals.

Executive Module

This module will tie in all the other modules and make them part of an integrated system that is easy to use and change.

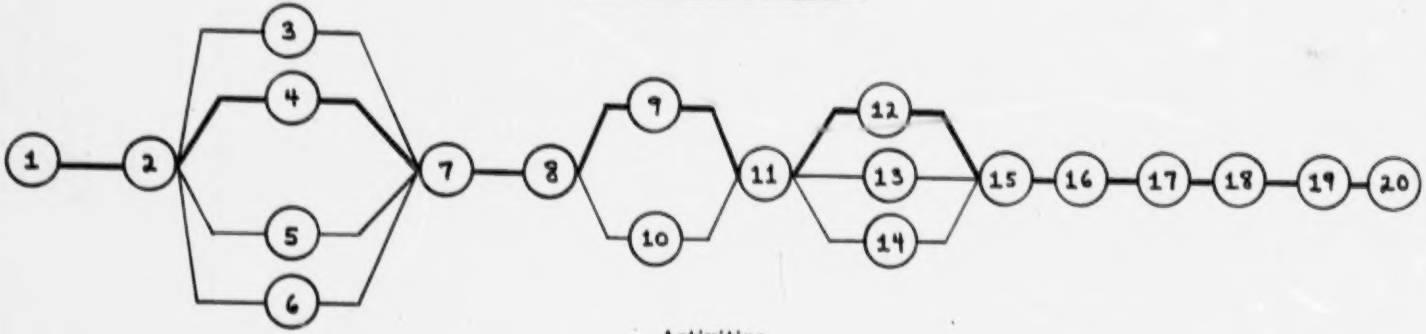
EXHIBIT 2

Group #1 IBM Project,

CPM Chart

Spring 1971, UCLA

Critical Path = 42 Days



Activities

1. Begin project
2. Establish initial IBM concept
3. Build data bases
4. Design operational control module
5. Design management control module
6. Design strategic planning module
7. Consolidate modules into IBM concept
8. Resolve discrepancies
9. Initial logic of interactive linkage
10. Enrich initial concept
11. Finalize IBM concept
12. Finalize operational control module
13. Finalize management control module
14. Finalize strategic planning module
15. Consolidate modules
16. Finalize interactive linkage
17. Test IBM
18. Debug model
19. Prepare final report
20. Turn-in report

Of the many interesting projects, the following one was selected as it is representative of an early project and includes a rather descriptive executive module.[9] Upon initiation the following (Exhibit 3) is received by the user.

EXHIBIT 3

```
THIS IS AN INTERACTIVE BUDGET MODEL FOR RISK INC.
THE MODEL DOES MANY MAGICAL AND EROTIC THINGS, AS FOLLOWS:

PRIOR YEAR BALANCE SHEET: TYPE BAL70
PRIOR YEAR INCOME STATEMENT: TYPE INCOME70
CURRENT DATA:
   TO ANALYZE CURRENT OPERATIONS YOU MUST SUPPLY CERTAIN VARIABLES.
THIS ALLOWS YOU TO ENTER WHAT YOU THINK THE ACTUAL FIGURES ARE, OR
ENTER THE POTENTIAL VALUES.  IT IS SUGGESTED THAT YOU BEGIN BY ENTERING
THE ACTUAL VALUES AND THEN VARYING ELEMENTS AS DESIRED TO OBSERVE THE
EFFECT ON THE OVERALL PICTURE.
THIS MODEL IS DIVIDED INTO THREE BASIC MODULES:
   OPC - THIS MODULE CONCERNS PRODUCTION ELEMENTS AND EFFECTS ON COST
OF GOODS SOLD.  TYPE OPC FOR THIS MODULE ALONE.
   MC - THIS MODULE PROVIDES BALANCE SHEETS, INCOME STATEMENTS, SALES
FORECASTS, ETC.  TYPE MC FOR THIS MODULE ALONE.
   SP - THIS MODULE CONCERNS STRATEGIC PLANNING AND PROVIDES LONG RANGE
PLANNING TOOLS.  TYPE SP FOR THIS MODULE ALONE.

EACH MODULE WILL DESCRIBE ITS VARIOUS FEATURES FOR YOU WHEN CALLED.

IN ADDITION, YOU CAN GET A BROAD OVERVIEW BY TYPING TOT.  USING THIS
YOU CAN FOR EXAMPLE VARY PRODUCTION COSTS AND IMMEDIATELY SEE THE EFFECTS
ON INCOME WITHOUT HAVING TO CALL EACH MODULE SEPARATELY.

YOU MAY BEGIN NOW -- HAVE FUN.
```

Given a general overview of the system the user is ready to begin. For the neophyte, historical financial statements are available such as last year's income statement:

```
SALES                       2700400
                          ------------

COST OF GOODS SOLD          1326000
SELLING, ADMIN EXPENSE      481000
INTEREST EXPENSE            13000
EARNINGS                    880400
TAXES ON INCOME             396180
                          ------------

NET INCOME                  484220
                          ============
```

Note that formatting options were not available to this group and thus the units in the preceding statement do not line up. To access the operating modules of this system, further documentation is needed. The various modules were accessed as displayed in Exhibit 4.

---

[9] Gordon, K.A., S. Archuleta, M. Ishii, "The Interactive Budget Model: A Conceptual Study," Term Project, Professor T. J. Mock, Graduate School of Management, University of California, Los Angeles, June 1971.

EXHIBIT 4

OPERATIONAL CONTROL:

```
      OPC
YOU HAVE ACCESSED THE OPERATIONAL CONTROL MODULE.
THIS MODULES FUNCTION IS TO ASSIST IN PRODUCTION PLANNING
USING A LINEAR PROGRAMMING ROUTINE TO MAXIMIZE A PROFIT FUNCTION.
YOU MAY VARY FACTORS OF PRODUCTION SUCH AS LABOR HOURS AND RAW MATERIALS
AVAILABLE.  THE PROGRAM ALSO INTERACTS WITH THE LONG RANGE SALES FORECAST
TO PROJECT PRODUCTION COSTS BASED ON PROJECTED SALES.

TO OPERATE THIS MODULE TYPE SAMCOST
      SAMCOST
THIS PROGRAM WILL CALCULATE THE OPTIMAL COMBINATION OF PRODUCTION
RESOURCES USED IN THE MANUFACTURING PROCESS OF THREE TELEVISION
MODELS:  X1=BLACK AND WHITE, X2=PORTABLE COLOR, X3=DELUX COLOR.
THE OBJECTIVE FUNCTION IS TO MAXIMIZE P=19X1+25X2+27X3, WHERE P
IS THE TOTAL CONTRIBUTION TO OVERHEAD AND PROFIT, AND X1, X2, AND
X3 ARE THE NUMBER OF UNITS TO BE PRODUCED TO MEET THIS OBJECTIVE.
THE OBJECTIVE FUNCTION IS BASED ON THE FOLLOWING DATA:
                                                X1    X2    X3
    SELLING PRICE PER UNIT                      90    350   425
    VARIABLE COST PER UNIT                      71    325   398
    CONTRIBUTION TO OVERHEAD AND PROFIT/UNIT    19    25    27
CONSTRAINTS ARE BASED ON THE FOLLOWING DATA
WHICH INDICATES THE AMOUNT REQUIRED
TO PRODUCE ONE UNIT:
     LABOR A                                    5     5     3
     LABOR B                                    8     12    14
     LABOR C                                    12    14    14
     MATERIAL D                                 10    12    12
     MATERIAL E                                 15    18    20
     MATERIAL F                                 1     1     1
INPUT PARAMETERS G, H, I, J, K, AND L, THE TOTAL LABOR HOURS
AND MATERIAL UNITS AVAILABLE.
⎕:
      10000
⎕:
      18000
⎕:
      24000
⎕:
      21000
⎕:
      25000
⎕:
      20000
        o
        o
        o
```

EXHIBIT 4 (CONTINUED)

MANAGEMENT CONTROL:

```
      MC
YOU HAVE ACCESSED THE MANAGEMENT CONTROL MODULE.
MODULE FEATURES ARE;
1.   CONSOLIDATED BALANCE SHEET - IN ORDER TO GET CURRENT STATUS
      TYPE BALCUR.  IF YOU WANT TO ENTER VARIABLESTYPE BALVAR.
2.   INCOME STATEMENT - IN ORDER TO GET CURRENT STATUS TYPE INCCUR.
     IF YOU WANT TO ENTER VARIABLES TYPE INCVAR.
3.   SALES FORECAST - TYPE FORE AND YOU WILL BE ASKED TO ENTER PROJECTED
     GROWTH RATE AND ACCEPTABLE STANDARD DEVIATION.  THE PROGRAM WILL THEN
     PROJECT MONTHLY DOLLAR SALES FOR THE PERIOD YOU SPECIFY AND PROVIDE
     YEARLY TOTALS.  YOU MAY THEN SPECIFY PERCENTAGE OF SALES BY MODEL AND
     GET SALES TOTALS BY MODELS, INCLUDING QUANTITY OF EACH MODEL SOLD.
4.   CURRENT RATIO - TYPE CR.
5.   ACID TEST RATIO - TYPE AT.
6.   RETURN ON ASSETS - TYPE ROA.
7.   EARNINGS ON STOCK - TYPE ES.
```

FORE ROUTINE (SALES FORECAST) IN MC MODULE:

```
      FORE
SALES FOR 1970 WERE 2,700,400.  FORTY-FIVE PERCENT WERE PORTABLE COLOR,
THIRTY-FIVE PERCENT WERE DELUXE COLOR, AND TWENTY PERCENT WERE BLACK AND WHITE.
ENTER BELOW FIGURES FOR PROJECTED SALES GROWTH (PERCENT YEARLY), ALLOWABLE
DEVIATION (FOUR FIGURE INTEGER), AND LENGTH OF FORECAST IN YEARS.
PROJECTED GROWTH RATE:
⎕:
      .05
DEVIATION:
⎕:
      1000
FORECAST PERIOD (INTEGER REPRESENTING YEARS):
⎕:
      2
                              SALES FORECAST
                    (BASED ON 5 PERCENT SALES GROWTH)

YEAR : 1
JAN  236290
FEB  236302
MAR  236105
APR  236772
MAY  236279
JUN  236052
JUL  235876
AUG  236733
SEP  235859
OCT  236286
NOV  236170
DEC  236063
TOTAL SALES: 2834787
```

EXHIBIT 4 (CONTINUED)

STRATEGIC PLANNING:

```
      SP
YOU HAVE ACCESSED THE STRATEGIC PLANNING MODULE.
THIS MODULE OFFERS LONG RANGE PLANNING TOOLS FOR THE FIRM.
MODULE FEATURES ARE:
1.  PRESENT VALUE OF INVESTMENTS - TYPE ISHII3SP1

2.  LONG RANGE SALES FORECAST (BASED ON GNP) - TYPE ISHII4SP2

THESE FEATURES WILL BE DESCRIBED FOR YOU WHEN CALLED.
```

OVERVIEW OF TOTAL SYSTEM:

```
      TOT
THIS MODULE OFFERS A TYPE OF OVERVIEW OF THE SYSTEM
THROUGH USE OF THE THREE BASIC MODULES.  IT IS USED
TO COMPARE THE LONG RANGE SALES FORECAST BASED ON GNP
AND THE FORECAST BASED ON PERCENTAGE SALES GROWTH.  THE
RESULTS OF THESE FORECASTS ARE THEN USED IN THE PRODUCTION
PLANNING MODULE, AND FINALLY A PROJECTED INCOME STATEMENT IS
PRODUCED.

TO OPERATE THIS MODULE TYPE TOT1
```

Contrasting the above first generation IBM with a third generation (class) results in some interesting insights, particularly in terms of system sophistication and improved output features. Consider an IBM designed to aid the financial planning, scheduling and control of the production of a motion picture.[10] The essentials of this project centered around a PERT/Cost schedule of the activities. The system included budget projections and critical path (Exhibits 5 and 6), an updating routine and subsequent project performance reports.

---

[10] Tompkins, G. E., H. D. VanHolth, S. E. Velasco, J.C.G. Gaspar and K. D. Prado, "An Interactive Budgeting Model for Producing a Motion Picture," Term Project, Professor T. J. Mock, Graduate School of Management, University of California, Los Angeles, Spring 1972.

EXHIBIT 5

Sample Budget Projections
(for "The Brazilian Connection")

TITLE: THE BRAZILIAN CONNECTION
DATE OF REPORT: 06/10/1972
INITIAL BUDGET

ACTIVITY CODE	ACTIVITY NAME	UNIT PRICE (DOLLARS)	QUANTITY	DURATION (DAYS)	SUB-TOTAL 2 (DOLLARS)	SUB-TOTAL 1 (DOLLARS)	TOTAL (DOLLARS)
1.1.0	STORY PURCHASE	100,000.00	1.0	0.0		100,000.00	
1.2.0	SCRIPT WRITER	100.00	1.0	15.0		1,500.00	
1.0.0	STORY AND SCRIPT						101,500.00
3.0.0	DIRECTOR	300.00	1.0	120.0			36,000.00
4.1.0	CAST	200.00	5.0	60.0		60,000.00	
4.2.0	EXTRAS	20.00	10.0	20.0		4,000.00	
4.0.0	CAST AND EXTRAS						64,000.00
6.1.1	DIRECTOR PHOTOGRAPHY	100.00	1.0	80.0	8,000.00		
6.1.2	OPERATORS	30.00	5.0	80.0	12,000.00		
6.1.0	CAMERA					20,000.00	
6.2.1	RECORDIST	30.00	2.0	50.0	3,000.00		
6.2.0	SOUND					3,000.00	
6.3.2	ELECTRICIANS	30.00	2.0	80.0	4,800.00		
6.3.0	LIGHTING, ELECTRICAL					4,800.00	
6.4.1	MAKE-UP,HAIR DRESSERS	40.00	2.0	60.0	4,800.00		
6.4.0	MAKE-UP,HAIR DRESSING					4,800.00	
6.7.1	EDITOR	100.00	1.0	30.0	3,000.00		
6.7.2	NEGATIVE CUTTER	30.00	3.0	30.0	2,700.00		
6.7.3	MUSIC, SOUND EDITOR	75.00	1.0	10.0	750.00		
6.7.0	EDITORIAL					6,450.00	
6.0.0	OPERATING STAFF, CREW						39,050.00
7.1.1	CAMERA EQUIP. RENTALS	10,000.00	5.0	0.0	50,000.00		
7.1.0	CAMERA					50,000.00	
7.2.0	SOUND EQUIP. RENTAL	5,000.00	2.0	0.0		10,000.00	
7.3.1	LIGHTING RENTAL	3,000.00	5.0	0.0	15,000.00		
7.3.0	LIGHTING RENTAL					15,000.00	
7.4.0	MAKE-UP,HAIR DRES.EXP	2,000.00	1.0	0.0		2,000.00	
7.6.0	EDITORIAL EXPENCES	10,000.00	1.0	0.0		10,000.00	
7.0.0	OPERATING EXPENCES						87,000.00
8.0.0	LABORATORY EXPENCES	5,000.00	1.0	0.0			5,000.00
9.0.0	MUSIC EXPENCES	3,000.00	1.0	0.0			3,000.00
						BUDGET TOTAL	335,550.00

EXHIBIT 6

Initial Critical Path for "The Brazilian Connection"

```
THE BRAZILIAN CONNECTION

BEGIN OF PROJECT  06/10/72
END OF PROJECT    10/22/72

                                                                    LAST DATES
ACTIVITY    PRECEDING ACTIVITIES         DURATION    SLACK      BEGIN     FINISH

  120                                       15                06/10/72  06/24/72    CRITICAL
  300        120                           120                06/25/72  10/22/72    CRITICAL
  410        120                            60         20     07/15/72  09/12/72
  420        120                            20         60     08/24/72  09/12/72
  611        120                            80                06/25/72  09/12/72    CRITICAL
  612        120                            80                06/25/72  09/12/72    CRITICAL
  621        120                            50         30     07/25/72  09/12/72
  632        120                            80                06/25/72  09/12/72    CRITICAL
  641        120                            60         20     07/15/72  09/12/72
  671        410  420  611  612  621        30                09/13/72  10/12/72    CRITICAL
             632  641
  672        410  420  611  612  621        30                09/13/72  10/12/72    CRITICAL
             632  641
  673        671  672                       10                10/13/72  10/22/72    CRITICAL
                                                                    LAST DATES
* PATH      PERTAINING ACTIVITIES        DURATION    SLACK      BEGIN     FINISH

    1        120  300                      135                06/10/72  10/22/72    CRITICAL
    2        120  410  671  673            115         20     06/30/72  10/22/72
    3        120  410  672  673            115         20     06/30/72  10/22/72
    4        120  420  671  673             75         60     08/09/72  10/22/72
    5        120  420  672  673             75         60     08/09/72  10/22/72
    6        120  611  671  673            135                06/10/72  10/22/72    CRITICAL
    7        120  611  672  673            135                06/10/72  10/22/72    CRITICAL
    8        120  612  671  673            135                06/10/72  10/22/72    CRITICAL
    9        120  612  672  673            135                06/10/72  10/22/72    CRITICAL
   10        120  621  671  673            105         30     07/10/72  10/22/72
   11        120  621  672  673            105         30     07/10/72  10/22/72
   12        120  632  671  673            135                06/10/72  10/22/72    CRITICAL
   13        120  632  672  673            135                06/10/72  10/22/72    CRITICAL
   14        120  641  671  673            115         20     06/30/72  10/22/72
   15        120  641  672  673            115         20     06/30/72  10/22/72
```

Results and Student Feedback

Two sources of data and experiences exist from which the pedagogical effectiveness of the IBM concept may be evaluated. First is the instructor's longitudinal observations over the three classes and comparison with previous courses. From this perspective the IBM simulation seems to be an improved educational aid for the various reasons given in the text of this paper including most importantly:

1. Its experiential nature
2. Reliance on group projects which reflect needed interpersonal and interdisciplinary approaches to system design
3. Replication of real world complexity, time pressures and technical difficulties
4. Ability to obtain an overview and understanding of the entire budgeting system.

Another source of feedback was student course evaluations that were collected in two of the courses. Although such appraisals are the result of a single exposure to the course and although student opinions are anything but consistent, the following critical and supportive quotes do lend positive evidence as to the overall usefulness and difficulties of the IBM in MIS education:

Critical

"Learned nothing about systems - only APL"

"Too much time was required with respect to coding"

"The IBM project was trying to cover too much material"

"Students tend to underestimate the effort involved in such a project"

"I learned nothing about planning and control ... too much time was wasted on the IBM"

Supportive

"IBM was creative and practical application of course subject matter"

"The IBM ... proved to be quite a meaningful education experience"

"Contributed much to my understanding of the budgeting process"

"An enjoyable learning experience"

"I especially liked working on the project although it does become time-consuming"

"Great learning experience"

"IBM was an excellent idea"

"The IBM was a demanding and interesting experience ... most of the (course) concepts acquired could be used in its design"

THE TRAFFIC POLICE MANAGEMENT TRAINING GAME

Gay Doreen Serway, Ph.D. Candidate
Allen S. Kennedy, M.S.
Gustave Rath, Ph.D.

Design and Development Center
Northwestern University
Evanston, Illinois 60201

Abstract

The Traffic Police Management Training Game was designed for Northwestern's Traffic Institute with the following basic objectives: (1) to provide police officers of supervisory rank with more insight and experience in traffic problems; (2) to show the importance of intensive analysis and planning; and, (3) to teach certain patrol enforcement concepts. The game requires three ingredients: the game administrators, the game players, and the computerized game model. The game model provides the framework within which the administrator may specify any type of urban model he believes meets his teaching objectives. The game players input decisions on allocation of manpower for patrol enforcement. The game model generates violations, which stochastically result in accidents. The frequency of these violations is assumed to be a function of parameters selected by the administrators and the enforcement applied by the players. If a violation does not result in an accident, the model computes whether or not an available unit detected the violation. The model does not pretend to be realistic, but rather aims to achieve verisimilitude. The object of the game is to achieve some "best" allocation based on criteria set up by the administrators. Organizational aspects of the game include: administrator and player briefings, instructions, decision forms, and critiques.

1.0 Introduction -- An Overview of the Game

The Traffic Institute of Northwestern University trains police officers from all over the world in the most modern techniques of traffic police administration. The Traffic Police Management Training Game was designed to integrate the principles, tools, and techniques taught in the Institue's program. The object of the game is to achieve some "best" allocation of the traffic police department's manpower and resources, based on measures of performance such as a reduction in the number of accidents, an increase in the detection of violations, etc.

The remainder of this section will be an introductory overview of the Traffic Police Management Training Game. The next section of this paper presents a basic definition of a management training game along with general objectives and characteristics of gaming. The Traffic Institute and the integration of its objectives into a training game are next delineated. The specifics of the Traffic Police Management training Game will then be discussed. The game model and simulation, administrator and player input, and the output of the game will each be presented. Some final notes from an actual experimental game session at the Institute will complete the paper.

Basically, the Traffic Police Management Training Game provides an opportunity for the player to formulate traffic control policies, allocate available resources and evaluate the results of his decisions within the context of a responsive gaming situation which is designed to react, or at least appear to react, realistically to his command decisions. Although the underlying relationships, as presented in the model are relatively simple, the total appearance of the game presents the players with a complex system.

The game model provides the framework within which the administrator may specify any type of urban traffic environment he believes best meets his teaching objectives. In addition, the administrator specifies the length of the simulated time period the game is to operate and the different daily patterns within the period. The basic unit of time assumed by the model is one hour, but the minimum length of an operating period is one day. Depending on how the game is to be used, it may be run for a simulated day, several days, a week, a month, etc.

The administrator is free to divide the class into several groups, each dealing with distinct urban systems, or to allow the class as a whole to "play" against one larger system. Once the urban system(s) have been specified, the administrator gives the class a city map, a brief demographic sketch, periods of operation, daily patterns, and whatever additional information about the city he may deem necessary.

The players must now organize their traffic departments for effective deployment of traffic units. This may include establishing line command structures, dividing the city into districts, etc. Having done this, the players are required to make initial manpower allocation for each daily pattern in the decision period.

Both the administrator and player forms are now keypunched onto computer cards in a prescribed format. This "input" card deck together with the actual program deck are submitted to the computer for processing.

The game model produces traffic violations with the simulated urban environment over time. The frequency of violations at any particular intersection is a function of the parameters inputed by the administrators and the enforcement applied by the players. Accidents are assumed to be the result of violations. When an accident occurs the closest (in time to respond) available unit is assigned to respond. If an accident has not occurred, the model computes whether or not an available unit observed the violation. If the answer is yes, a detection has occurred. Thus, the probability of detecting violators increases with an increased manpower allocation.

The object of the game is to achieve some "best" allocation based on the criteria the players select to measure performance. Each play of the game generates data in report form for analysis and use in making decisions in subsequent plays of the game.

The next section of this paper presents the basic management game theory which underlies the Traffic Police Management Training Game.

2.0 Management Gaming Background

2.1 Definition

> Management games are "games" in the sense that there are participants, a set of rules, and a method of scoring.[15]

A management game consists of four elements. There are the players who assume roles in an organization. There is the model which simulates the environment in which the organization operates. There is the input which consists of decisions by the players and there is the output which is generated by the model and provides feedback to the players from which to "score" the results of their decisions.

A management game puts players into a simulated environment where they assume roles in an organization characteristic of this environment. In their roles as administrators, the players make decisions applying their experience and knowledge to achieve certain objectives for the organization. Through decision-making they become aware of the interrelatedness of administrative actions in the organization.

When a decision is made the simulated model of the environment uses the decisions as inputs and generates changes in the conditions of the environment. Reports are produced by the model and the players now must make new decisions using this feedback from the game. The player is, thus, actually living with the consequences of his decisions. Several plays of the game may simulate a year's operation of the organization. The player, thus, has an opportunity to make decisions, see their results but not suffer the real world consequences, such as, bankruptcy, war, or famine.

2.2 The Objectives of Management Games

Searching through the literature on management games a list of objectives of management games in education can be compiled. While any list would be only partial, we have found the following general objectives:

1. to provide a dynamic, reacting time dimension
2. to provide objective feedback
3. to provide an opportunity to learn from experience
4. to provide an opportunity for experimenting with different decisions
5. to provide an opportunity for an overview of the organization
6. to provide an opportunity to integrate knowledge and experience
7. to provide an opportunity to learn to work together

A brief description of each of these follows.

Dynamic

Kibbee has said,

> The two unique characteristics which enable games to contribute so powerfully to management education are the novel use of the time dimension, and the objectivity of the feedback. [15]

The management game is virtually alive. Its state is constantly changing in response to decisions. Further, the management game condenses a large amount of decision-making experience into a relatively short period of time. As Thorelli has said about business management games:

> While a case study of the traditional type provides an essentially static snapshot of a business problem situtation, a game yields a moving, multi-dimensional picture. [22]

Feedback

Objective feedback is provided

> in a mangement game by a set of programmed relationships which transform the input decisions into performance reports. It enables the participant to analyze the actual responses of a business environment. [15]

The objective of providing feedback is linked closely with the dynamic aspect of management games. When the player makes his decision, the management game proceeds to implement it, the simulated environment changing in response to the players' inputs. The player, thus, has the results of the application of decisions. In effect, business management games, thus, are like "case studies with feedback and a time dimension added". [31]

Absence of real-world consequences

Closely linked with the dynamic time dimension is the management game objective of providing an opportunity to learn from experience without paying the price that would result from wrong decisions made in real life, for example, being fired.

As Kibbee points out:

> The player is learning by implementing decisions without disrupting established operations, incurring the cost of mistakes, or inviting the resistance of vested interest. [15]

Experimentation

Management games can make experimentation possible, because it is always possible to return to a previous point in the simulation and proceed again from that point, making a different set of decisions to determine their advantages and disadvantages in comparison with those previously tried. Clarkson College, [15] put this feature of management games to a unique use by permitting those participants who said, "I wish I had it to do all over again", to do it all over gain, by resuming the exercise at the point where they feel they went wrong.

Overview

A vital purpose of management games in education is to provide the players with an overall perspective of their organization and to improve their feel for the interrelatedness of the various functions. By dynamic role playing, the player is forced to think about the interrelated aspects of functions and responsibilities. The players also become aware of the interrelatedness of short-and-long range planning for the successful operation of their organization.

Other objectives

Several other objectives are claimed for management games in education. Management games offer an opportunity for applying and testing knowledge gained from reading and other experiences.

The players can become personally involved in a simulated situation and find ways to work together under pressure in developing their decision-making abilities.

One final purpose for using management games, when programmed for a computer, is that the games are often a good way to introduce management to the realm of electronic data-processing equipment and computers. [15]

2.3 Characteristics of Management Games

Management games exhibit a degree of polarity in the individual characteristics which can be developed in the game. An enumeration of these includes:

Simple	Complex
Manual	Computerized
Deterministic	Stochastic
Functional	Total organization
Non-interactive	Interactive
Qualitative factors included	Only quantitive factors
Use for single play	Use for repeated plays
Discontinuous play	Continuous play

Verisimilitude

Special features

Simple or Complex?

Complexity in a management game may manifest itself in the game rules, in the structure of the decision forms, in the simulation model itself, in the number of decisions to be made and so on. In a complex game, "undue anchorage of the model in the details of a specific industry, or specific parts of the world, may cause disputes about institutional facts and relationships of no real consequence to the objectives of the game and very well may divert the attention of the participants to peripheral matters". [22]

A review of the literature clearly indicates that range from the depth and complexity of the Carnegie Institute of Technology Game, in which as many as 300 decisions may be made each "month" to the simplicity of the original American Management Association's Game, in which only six decisions were required.

The question of whether simple management games can be used to illustrate management principles has been frequently asked. In a study of the educational value of management games by Anthony Raia [51], the hypothesis that a relatively simple game provides essentially the same benefits as one that is more complex, in terms of learning, attitude, and levels of interest and motivation was accepted. [51]

The choice of a simple or a complex management game rests primarily upon the particular objective of using a management game in a course, seminar or wherever.

Manual or Computer

Management games can be classified as manual or computer on the basis of how the computations, required to convert the decisions made by the players into the performance reports returned to them, are made.

Kibbee defines a manual game as one "in which the computations are made by clerks, or by the participants themselves, usually with the help of desk calculators". [15] Similarly, computer games are ones in which the computations are performed by electronic data processing equipment, analog and digital computers.

Greenlaw states:

> Both approaches (manual and computer) to the computation of decision results have their advantages and limitations. Although complexity can be more easily incorporated into computer-designed models, results calculated more quickly, and more comprehensive reports produced, manual calculation is much less expensive and permits more flexible game administration. [8]

The basic factors involved in deciding upon computer or manual methods for management game computations have been discussed by Greenlaw [81], Kibbee [15], thorelli [22], and others. The factors include:

1. speed and complexity of the model
2. accuracy required
3. cost
4. flexibility
5. reports generated
6. ease of experimentation
7. glamour

Deterministic or Stochastic

In a management game, the mathematical model is the set of relationships from which the output report is computed from the input decisions. Some management games are completely deterministic models--models in which operating results are determined solely by the decisions made by the players, and not by chance. Others, utilize stochastic or probabilistic models, in which chance plays a role in one way or another and influences the outcome of the game. A business game with stochastic elements, for example, may put certain variables, such as the fluctuations in the general level of business activity beyond the control of the players. Games are usually a mixture and seldom purely stochastic. [23]

Functional or total organization

"A general management game is designed to teach decision-making at the top management level where all major functional areas of the total organization are involved in achieving fundamental organizational objectives."[31] In the business management games, called total enterprise games, the basic problem is the management of a complete company. Typical decisions to be made in total enterprise games include:

1. price of product
2. marketing budget
3. research and development budget
4. maintenance budget
5. production volume scheduled
6. investment in plant and equipment
7. purchase of materials
8. dividends declared

Functional games are intended to teach specific skills in a particular management area. These games are highly specialized, confined largely to problems within a relatively narrow area. Functional business games may deal specifically with:

marketing
maintenance management
material flow and inventory
production scheduling
personnel
physical distribution
toolroom operation
manufacturing scheduling
finance, asset management
procurement and supply
salesmanship
and so on [8]

Interactive or non-interactive

According to Greenlaw, a game is classed as interactive "if the decisions made by one group of participants have a specific mathematically determinable effect upon the results achieved by other groups of participants". [8]

Kibbee has stated this characteristic very interestingly:

> A game with interaction is like tennis; a game without interaction is like golf. [15]

Further, he gives examples from business management games.

> In the Univac Marketing Game the various teams are in competition for a common market, and the action of any one team, say in its pricing policy will affect all the other teams; there is an interaction between teams. In the Westinghouse Inventory Control

> Game, on the other hand, each team is attempting to achieve the best performance beginning from the same conditions, but there is no interaction between teams; the performance of one team has no effect on the other teams. [15]

Qualitative or only quantitative factors

In a management game, the relationship between a particular decision and its effect usually results from computations performed using mathematical relationships built into the game model. It is possible, however, to use human beings and introduce qualitative factors, their "rational" judgements as to the results that should ensue on a particular decision. These people are called judges, referees, or umpires. [8]

Another type of qualitative factor occurs in the information flow in management games. The management game may generate standard reports from inputs or may include a qualitative decision as to the quantity and quality of reports the players would like to have. The players decide upon what information they will buy from the available quantity of data.

Single or repeated plays

Management games are played in "periods". This interval of time, e.g., a week, a month, a quarter, a year, is called the simulated time period and decisions are made for the length of a period of play. Decisions are made in real time for the simulated time period. Management games may, thus, last in real time for an hour or a month, while the simulated time period may be a quarter or a year.

In a single play, the players make their decisions for the simulated time period, receive the outcome, and the exercise is over; the feedback from decision-making cycle is two step. In repeated plays, decisions are made, feedback results, new decisions are made with the feedback contributing to decision-making, more feedback, and the cycle goes on.

Discontinuous or continous play

Decision-making in management games may be continuous or discontinuous. In continuous decision-making game sessions, decisions are made at one sitting, uninterrupted by other work or adjournments. Discontinuous decision-making takes place over many days or weeks. Usually in discontinuous sessions, the management game is an adjunct or a complement to other course or training activities, rather than being the only activity, which is the case in continuous play.

Verisimilitude

Verisimilitude is the appearance of reality to the player, but it does not imply realism of the model. Complexity is not necessary for verisimilitude, the desired effect can be easily achieved by very simple models. [15]

As Kibbee points out,

> When a game is to be used solely for training purposes, it is not necessarily important that the mathematical model be realistic, but only that the game appear realistic to the players. The appearance of realism is known as verisimilitude, and this is an essential feature of management games. The men who play management games react to the business simulations very much as they do in real life. [15]

Using an adapted version of the game outline developed by Greenlaw, et al. the Traffic Police Management Training Game can be summarized in the terminology of this section, as follow:

I. Organization: Northwestern Univeristy Traffic Institute

II. Name of game: Traffic Police Management Training Game

III. Characteristics of the model:
   A. Specific: deals only with traffic violations
   B. General view: Not functional, rather aims to interrelate organization, management, planning, statistical analysis, etc.
   C. Stochastic: Major events occur stochastically
   D. Non-interactive: Decisions of one team do not effect the other teams
   E. Entirely quantitative, except for administrator input

IV. Characteristics of the Administrative process:
   A. Role positions suggested but not assigned by the administrators
   B. Computer
   C. Use for repeated runs
   D. Discontinuous play

2.4 A procedural Point in Using Management Games -- How to Choose a Game

The method of systems analysis lends itself to selecting the appropriate management game for any organization. The first step is to set forth the educational objectives expected of the game. Then, certain criteria may be imposed, for example, the game must have discontinuous play. Next, establish any constraints on the game, such as the abilities and interests of the players and administrators. Now, determine the resources available for the program in terms of finances, time, staff, computing facilities, and such. On the basis of information from published material, professional organizations, or game builders on all games being considered, a list of alternatives can be proposed. The list of alternatives may include one specific game, a modification of an existing game, or the design of a new game. Now the administrators must select one alternative and schedule it in the overall program in such a way as to achieve maximum impact on the players. Finally, the administrators should apply the systems analysis approach to the game to see whether the teaching objectives were reached, and to learn how the program can be improved. [15] [18]

In the next section of this paper we will discuss the Traffic Institute and how its objectives lead to the development of the Traffic Police Management Training Game.

3. The Traffic Institute and Its Objectives

Police administrators across the United States, today, realize the need for command and supervisory officers capable of assuming responsibilities in the administrative function. These officers must understand the scope of the entire traffic accident prevention program, its coordination, and the interrelation of such programs. They must know the principles and techniques necessary to reduce the number and severity of motor vehicle collisions and how to cope with the problems of congestion created by more drivers and more vehicles.

The Traffic Institute of Northwestern University, through its nine-month Traffic Police Administration Training Program, provides specialized training that can prepare law enforcement officers to meet the demands brought about by today's conditions. Following the Northwestern University quarter system, the fall quarter is devoted to subjects of common interest with specialized study in the selected areas beginning in the winter and continued through the spring. The full program provides the student with a general education background as well as specialized study in selected areas such as traffic programs, management and training.

Municipal, county, and state police officers come to the Institute from all over the world sent by their home police departments. These police departments have recognized the need for having supervisory staff trained in the most modern concepts, tools, and techniques of traffic police administration. An officer who has completed the program typically returns to his home department to set up training programs, teaching the techniques he has learned at the Institute.

Analysis of the Traffic Institute's program yields the following general objectives:

1. to instill in each participant the knowledge to operate successfully

in and contribute significantly to the operation of the police organization of which he is a part;

2. to provide detailed instruction for students with special interests in the areas of police management, traffic administration or police training;
3. to provide an educational base in areas related to police work such as oral and written communications, sociology, law, etc.;
4. to provide the instruction in and the opportunity for independent research in areas of interest. [72]

The staff of the Traffic Institute expressed interest in using gaming techniques in their program, in the Fall of 1968. This interest stemmed from the wide acceptance business games had received in business management training programs. Business games could provide an atmosphere for applying general management principles. Thus, we could expect that such games would contribute to the management aspects of police work. We might expect a contribution in the following areas:

1. Principles of modern management
2. Organization
3. Planning
4. Analysis for decision-making.

During discussions that Fall with members of the Traffic Institute staff, a number of qualitative arguments were raised against the use of any existing business management game for police training purposes. Some of these were:

1. police training objectives are distinctly different from business training objectives (although some overlap of principles exists);
2. the real-life systems are radically different resulting in a loss of the highly important verisimilitude concept for police trainees participating in a business game;
3. unlike the police system, the business system is highly interactive.

In short, a business game is not directed toward police activities, so it would seem that a management game directed at police systems to serve police training objectives would be desirable. Such a game did not exist.

It was at this point that the original ideas for the Traffic Police Management Training Game were put forth. The game would be limited to the traffic department of a city police department. One objective of the game would be to focus the player's attention on the interrelationships between various methods of allocating policemen in a city to achieve a "best" effect in terms of accident reduction and violation detection. Other objectives would be to stimulate the player's thinking about organization, management, planning, and statistical analysis. The objective of the game would be to achieve a set of "normal" short and long range operating conditions within the department by applying sound police management principles. In this game, the emphasis would be on sound management decisions and planning based on the use of data analysis techniques. The game would contribute to the following objectives of the training program:

1. Principles of modern management
2. Organization
3. Planning
4. Analysis for decision-making
5. Traffic direction and control
6. Analysis and use of traffic records
7. Use of statistics and budgeting

In addition to contributing to these objectives, the designers hypothesized that the game experience for students who would later be training policemen in their own departments would provide an opportunity to evaluate the use of games in their own programs. At this point the game designers began to formulate a model of traffic to be used in the game.

4. The Model for the Game

The most significant problems associated with the operation of a traffic division are associated with the optimum allocation of the available resources of time, manpower, equipment, and budget to cover urban traffic control and accidents, and with the formulation of police policies for dealing with traffic flow situations, accidents, violators, issuance of citations, etc. The game model must, thus, establish the cause and effect relationships between allocation of manpower to tne various functions and the community response and reaction.

Kibbee [81] points out that the most important requirement in a management game model which is used solely for training is verisimilitude: The appearance of reality. Verisimilitude can be achieved with amazingly simple mathematical relationships. Although the underlying relationships may be relatively simple, the total appearance of the model presented to the players is that of a complex system. In this section the mathematical equations which describe the environmental response to enforcement allocations made by game participants are presented. [76]

The objective of the model is to simulate traffic violation occurrences in time and space. The model also keeps track of the movement of patrol units in their assigned areas. Detection of a violator requires the simultaneous presence of a non-busy patrol unit and a violation occurrence. In addition, the model simulates accidents and assigns the closest (in time, not space) patrol unit to respond. While the model is capable of generating many types of statistics, currently it generates the following:

1. detected violations, by type, location, time
2. accidents by time and location
3. response times
4. service times
5. unit assigned to accident or handling violation

To begin, a definition is required.

Definition--Enforcement cell (i,j)

By an enforcement cell (i,j) we shall mean the one half block in all directions from intersection (i,j). This concept is illustrated in Figure 1.

For discussion purposes, it is sufficient to consider the development of one hour n within the daily pattern p (a definition of daily pattern is given in the Input section). The model will range over each hour n for all days $d = 1,2...,N_p$ in pattern p and all patterns p.

Determining violation events

We shall assume that the number of violations occurring in cell (i,j) during hour n obeys a Poisson distribution with parameter $V_{ij}(n)$. The time t of occurrence of each violation is selected from a negative exponential distribution (n-1 to n). It remains to describe the variation in $V_{ij}(n)$ with enforcement and time.

Police enforcement

Assume that patrol unit k has been assigned to area $A_k(n)$ and will weight his time according to the intersection flow rates, $Q_{ij}(n)$, where $(i,j) \epsilon A_k(n)$. Thus, we assume that the probability of finding unit k in a particular cell (i,j) at time t during hour n is given by

$$P\left[k \epsilon (i,j) \epsilon A_k(n) \text{ at time } t\right] = \frac{Q_{ij}(n)}{\sum\limits_{(i,j) \epsilon A_k(n)} Q_{ij}(n)} . \quad (1)$$

Definition -- Enforcement level $E_{ij}(n)$

We define the enforcement level, $E_{ij}(n)$, in cell (i,j) during hour n as the sum of the probabilities of finding a patrol unit in cell (ij) at any time t during hour n. That is,

$$E_{ij}(n) = \sum_{k \epsilon K_{ij}(n)} P\left[k \epsilon (i,j) \epsilon A_k(n) \text{ at time } t\right]. \quad (2)$$

In particular, if only one unit k has been assigned to spot patrol at intersection (i,j), then

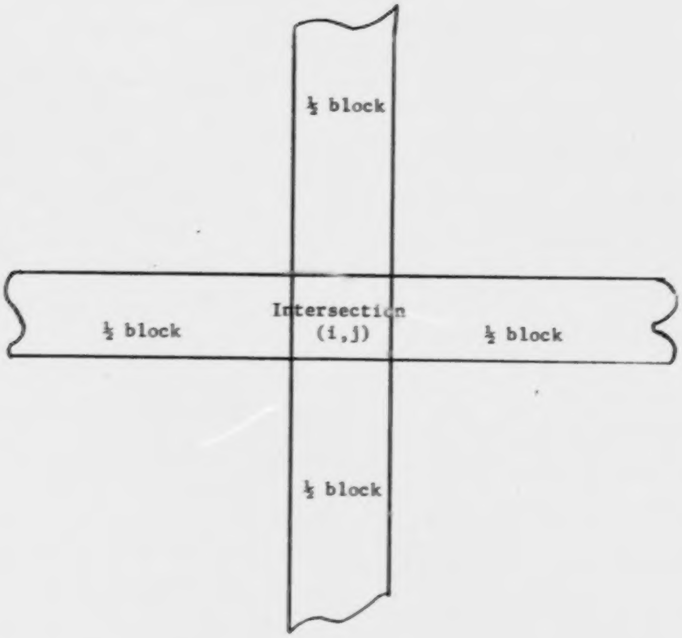
$$K_{ij}(n) = k \text{ , } A_k (n) = (i,j), \; p\left[k \epsilon (i,j) \text{ at } t\right] = \frac{Q_{ij}(n)}{Q_{ij}(n)} = 1,$$

and $E_{ij}(n) = 1$.

We are saying that applied police enforcement, i.e., number of units assigned to

Figure 1

Enforcement cell



cell (i,j), is the sum of the probability defined in (1) for all units assigned, for the same hour of each of the same pattern day types. Thus, there is an applied police enforcement for each cell (i,j), as a function of intersection flow rate, number of units assigned, and previous duplicate pattern days' enforcement at the same hour n.

Community perceived enforcement

Following Fisher and Mosher [6], we assume that the violation rate, $V_{ij}(n)$, will decay exponentially as a function of both enforcement and time. For the present we may drop the indices (i,j) and n.

First, we allow V to vary exponentially between $\bar{V}$ and $\underline{V}$ as follows:

$$V(E) = \bar{V} e^{-\gamma E} + \underline{V} \left[1 - e^{-\gamma E}\right] . \tag{3}$$

To determine the free parameter γ, let

$$V \quad V^E = \left[1 - \frac{K^E}{100}\right] \bar{V} , \tag{4}$$

and require for E = 1 that

$$V^E = \overline{V}\, e^{-\gamma} = \underline{V}\,[1 - e^{-\gamma}] \qquad (5)$$

$$= \underline{V} + [\overline{V} - \underline{V}\,]e^{-\gamma} \qquad (6)$$

or $$e^{-\gamma} = \frac{V^E - \underline{V}}{\overline{V} - \underline{V}} \qquad (7)$$

or $$\gamma = -\ln \frac{V^E - \underline{V}}{\overline{V} - \underline{V}} \quad . \qquad (8)$$

Note, that we have defined a "community" response to a change in enforcement. However, the neighborhood generally has to be exposed to a change in enforcement before the complete effect is realized.

To describe this "time delay" response, let $V^{(0)}$ be the violation rate at an old enforcement level $E^{(0)}$, and $V^{(1)}$ be the violation rate at the new enforcement level, $E^{(1)}$. Again, assume an exponential decay after d days exposure to the new level $E^{(1)}$ as follows:

$$V(d) = V^{(0)}e^{-\tau d} + V^{(1)}\ 1- e^{-\tau d} \qquad (9)$$
$$\text{where } V(0) = V^{(0)}.$$

To determine the free parameter τ, recall that N^E is the number of days of continuous enforcement a $E \underline{=} 1$ required to drop the violation rate from $\overline{V}$ to V^E. Thus, we require

$$V^E = \overline{V}\, e^{-\tau N^E} + \underline{V}\,[1 - e^{-\tau N^E}], \qquad (10)$$

or

$$\tau = -\ln \frac{\dfrac{V^E - \underline{V}}{\overline{V} - \underline{V}}}{N^E} \quad . \qquad (11)$$

Such curves describe how quickly (in terms of days) a community responds to a change in enforcement.

Generating the Master Violation Table (MVT)

Equations (3) and (9) allow for the computation of violation rates for all cells (i,j) for each hour n. The number of violations, N^V_{ij}, of each type occurring in each cell (i,j) during hour n, for each n, is now determined from a Poisson distribution with parameter $V_{ij}(n)$. For each cell (i,j) we select the violation occurrence times from a negative exponential distribution with parameter $1/\ V_{ij}(n)$.

For simulation purposes, we wish to string violation occurrences on a time line in minutes during one hour n. To accomplish this, we construct a master violation table (MVT) of violations ordered in ascending time of occurrence. The first entry in this MVT is the time of occurrence of violation, the second is type of violation, the third is avenue number where violation occurred and the fourth is street number where violation occurred. Once the MVT is sorted in time sequence, we are ready to run the simulation for hour n by sequentially processing each violation in the MVT.

Patrol unit location

In order to keep track of patrol units in space and time we shall assign to each unit the following attributes:

1. t_F -the time the unit is available for reassignment;
2. t_L -the time the unit was last located;
3. i_L -the avenue index when the unit was last located;
4. j_L -the street index when the unit was last located;
5. the current area assignment.

When attempting to locate a unit at time t, we encounter three cases:

1. $t_F > t$-the unit is busy;
2. $t_F < t$-unit is available and inside its area;
3. $t_F < t$-unit is available and outside its area.

For cases (2) and (3) we wish to compute the unit's new position (i,j) at the present time t.

For case (2), we assume the unit has been patrolling its area since last located at time t_L. However, if t_L is recent relative to t, it may not be reasonable to expect that the unit could be in every cell of $A_k(n)$. Thus, we require the following:

Definition--Region of influence $R_k(t)$

We define the region of influence of unit k at time t as that coordinate rectangle in which unit k can be realistically located. Let

i_R^A = right (east) boundary of the coordinate rectangle $A_k(n)$

i_L^A = left (west) boundary of the coordinate rectangle $A_k(n)$

j_T^A = top (north) boundary of the coordinate rectangle $A_k(n)$

j_B^A = bottom (south) boundary of the coordinate rectangle $A_k(n)$.

Similarly let,

i_R^R = right boundary of the coordinate rectangle $R_k(t)$

i_L^R = left boundary of the coordinate rectangle $R_k(t)$

j_T^R = top boundary of the coordinate rectangle $R_k(t)$

j_B^R = bottom boundary of the coordinate rectangle $R_k(t)$.

Assuming some average patrol speed, v_p, we can compute the bounds on $R_k(t)$ as follows:

$$i_R^R = \min\left[i_L + v_p\,(t-t_L),\ i_R^A\right]$$
$$i_R^R = \max\left[i_L - v_p\,(t-t_L),\ i_L^A\right]$$
$$j_T^R = \min\left[j_L + v_p\,(t-t_L),\ j_T^A\right]$$
$$j_B^R = \max\left[j_L - v_p\,(t-t_L),\ j_B^A\right]$$

This concept is illustrated in Figure 2.

We now select the current location of the unit from the probability function

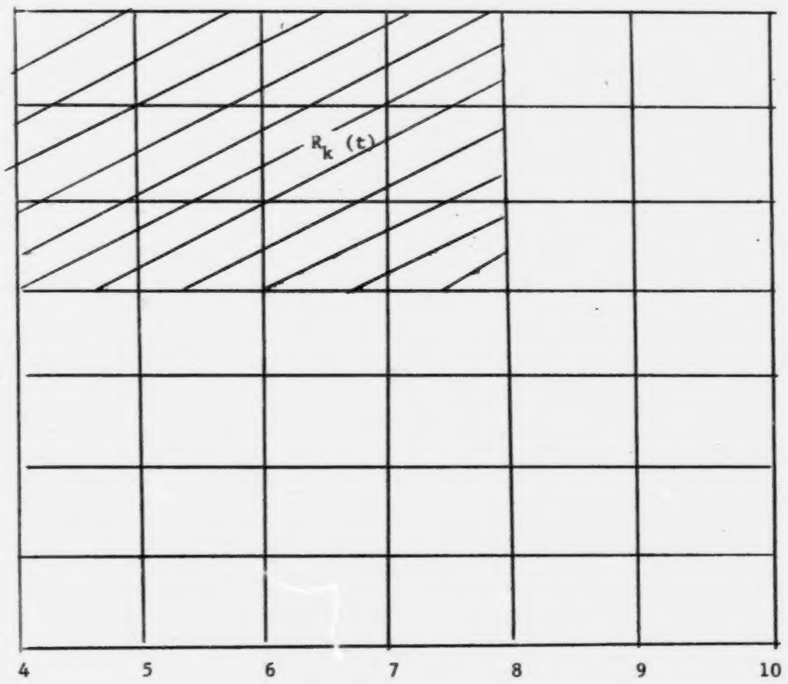
$$\sum_{(i,j)\,\epsilon\,R_k\,(t)} P\left[k\ \epsilon\ (i,j)\ \ \epsilon\ R_k(t)\right] = \frac{Q_{ij}(n)}{\sum\limits_{(i,j)\epsilon R_k(t)} Q_{ij}(n)} \qquad (12)$$

This probability for each $(i,j)\ \epsilon\ R_k(t)$ is compared to a random number from a uniform

Figure 2

Region of influence

$A_k(n)$



$R_k(t)$
19
18
17
16
15
14
13
12
4
5
6
7
8
9
10

distribution. When the probability in (12) is greater than the random number, this (i,j) becomes the cell where he is.

For case 3, unit is not busy, outside area, we assume the unit returns to its area by the shortest route possible. Here, we have three subcases as depicted in Figure 3. In each case, the model computes whether or not the unit has had a chance to return to its area. If not, it will compute the unit's position along the route of shortest return. If the unit has had time to return, its time and location of entrance into the area will be used to compute a region of influence and the current location will be selected as described previously.

Event accident

When a violation is selected from the top of the MVT list, the question of whether it resulted in an accident must be decided. The model assumes accidents happen with a uniform distribution. For each violation type v, a number is inputed (see the next chapter, Input section) representing the percentage of violations of type v that result in accidents. When a violation is selected from the MVT, the model checks whether the uniform number is greater than this percentage, indicating no accident occurred.

If an accident occurs then the model computes the response time for all non-busy units and assigns the unit with shortest response time to cover the accident. Service time is negative exponential with mean $1/S_v$, where S_v, average service time for an accident resulting from this violation is also inputed. The model updates the servicing unit's location, i_L, j_L, availability attribute, and time available for reassignment, t_F.

Event No Accident

If no accident occurs as the result of a violation, two further possibilities remain. Either the violation is detected or not detected. The model requires the unit to be at the same location where the violation occurs in order for the violation to be detected. The test for detection involves locating each unit, determining whether it is available or not and, if it is available, locating the unit, as described in a previous section, cases (2) and (3). Once located the probability of detection becomes:

$P = 0$ if $i_L, j_L \neq$ iviol, jviol and

$P = 1$ if $i_L, j_L =$ iviol, jviol, where iviol, jviol is the cell where violation occurs.

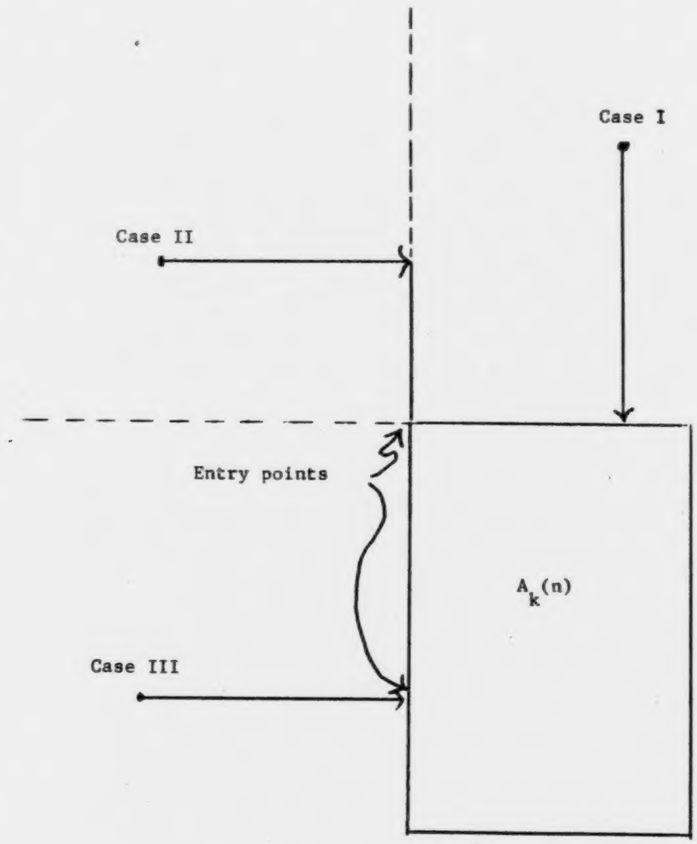
Model Logic

It is now possible to define the logical sequence of operations the model must execute.

1. Compute police enforcement
2. Compute community perceived police enforcement

Figure 3

Return Routes to Assigned Area


Case I
Case II
Entry points
$A_k(n)$
Case III

3. Generate Master Violation Table
4. Select violation from the top of the MVT
5. Did an accident occur?
    a) If yes, go to 6
    b) If no, go to 7
6. Locate all available units at time t and assign closest to respond to accident. Record response time, service time, accident, violation type and unit responding.
    Take unit out of service. Go to 4.
7. Locate available units $k \epsilon K_{ij}(n)$ and for each, test if unit location is (i,j).
    a) If no, test if $K_{ij}(n)$ is exhausted.
        i) If yes, go to 4, violation undetected
        ii) If no, continue search on units $k \epsilon K_{ij}(n)$
    b) If yes, record violation type, service time, unit responding.
        Take unit out of Service. Go to 4.
8. The end.

5. The Simulation

The simulation is written in Fortran IV and was developed for use on the CDC6400 digital computer at the Vogelback Computer Center on the Evanston campus of Northwestern University. The simulation is based on the model just presented.

Applying some of the terminology of simulations, the Police Traffic Management Training Game's simulation basically has the following entities, entity attributes, and events events.

	Entities	Attributes
Active:		
	Patrol unit	District to which it belongs Patrol speed Response speed Busy or not busy Inside or outside area Time of day it was last located Location (i,j) where last located Time of day when it will be non-busy The area to which it is assigned for the current hour and daily pattern
Passive:		
	Region	Avenue and street boundaries Qualtative attributes (i.e., ghetto,

	industrial, etc.)
Districts	Avenue and street boundaries Police Headquarters An allocation of patrol units
Intersection$_{(i,j)}$	Region in which it is located District in which it is located Area in which it is located Traffic flow rates for each hour of daily pattern

Events	Comments
Violation	The basic event generated by the Traffic System. It has the following characteristics: Type Average time to service a detection Average time to service an accident resulting from this type violation Maximum violation rate ($\bar{V}$) for this type Minimum violation rate ($\underline{V}$) for this type Percentage of this type of violation resulting in an accident
Accident	Requires a violation to occur first, the system generates accidents probabilistically
Detection	Requires a violation to occur first, the system generates non-accident violations which the structure defines as detected if the unit is available and at that intersection (i,j)

6.0 The Input

The Traffic Police Management Training Game involves two kinds of input, as indicated in the overview section. There is administrator input and player input.

6.1 Administrator Input

The administrator by inputing parameters defines the simulated urban environment in which decisions will be made by the players. Some of these parameters will be made known to the players, others are useful only to the model for generating violations, accidents, and so on, but would be of no value to the players even if known, for example, $\overline{V}$ and $\underline{V}$ for each v, since they represent subjective judgements for the administrators. Clearly, the administrator defines the reference city by his inputs for city parameters. He specifies whatever type of urban traffic environment he believes best meets his teaching objectives. Further, the administrator establishes the length of the simulated time period.

The following elements are required to specify the urban traffic system:

1. City limit definition
2. Violation table
3. Daily pattern table
4. Composition definition
5. Region definition
6. Composition to region assignments for each daily pattern.

A brief description of each of these follows:

City limit definition

The basic unit of geographic area is a rectangle whose sides are parallel to a coordinate axis system. The city, for the sake of simplicity, is thought of as one large coordinate rectangle. City boundaries are thus specified by the dimensions, in terms of avenues and streets of a rectangle. The city limit definition thus is designated "from avenue____ to avenue ______ and street ______ to street______". Figure 4 illustrates this terminology.

Violation table

The violation table is merely a list of all the violation types to be considered in game play. They are identified by an eight character code and a description of at most sixty characters, including blanks.

Daily pattern table

The daily pattern table introduces the time dimension into the game. The daily pattern table serves to define the different daily patterns which are to be considered and how many of each pattern are to be included in the play period. A daily pattern might be a particular day of the week, like Monday, or just the category weekday, or Saturday, or Sunday.

Composition definition

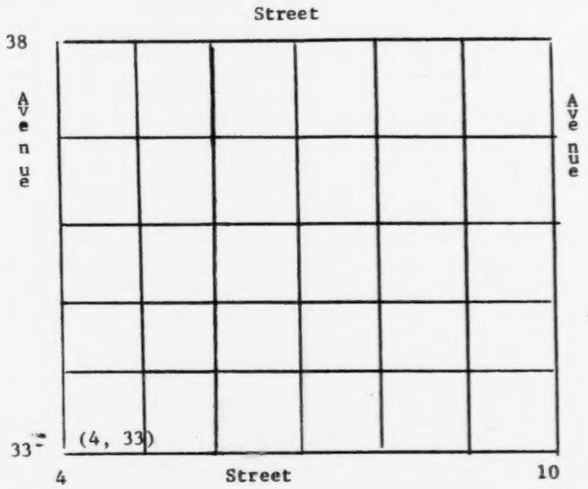
Game "parameters" are input in a block termed a "composition". A composition is an artifical device for assigning constant parameters over time and space. It is a particular set of parametric values that can be assigned to the time and space dimensions where appropriate.

Associated with each composition are the following parameters:

1. intersection flow rate in vehicles per hour
2. patrol speed (miles per hour)
3. response speed (miles per hour)
4. the percentage of vehicles which will

Figure 4

The Coordinate Rectangle City Limit Definition



be in violation if the drivers have the knowledge that they will be not observed or stopped by a patrol unit (i.e., zero level enforcement)

5. the percentage of vehicles which will be in violation if the drivers have the knowledge that they will be continuously observed and violators will be stopped by a patrol unit (i.e., unlimited enforcement)
6. the reduction in violations that can be expected, due to a single patrol unit being continuously assigned for a specified number of days at a particular intersection (defined as an enforcement level of unity)
7. the number of days a patrol unit must be assigned to achieve the reduction of (4) above (this parameter defines the time response to patrol)
8. the expected percentage of violations of this type resulting in accidents

These parameters are sufficient to give not only the physical flow rates, but also allow for a description of the community response to enforcement over time. A community with respect for traffic law will have a low violation rate, regardless of enforcement. A community with respect (or fear of traffic enforcement) will quickly respond to patrol activity. Problems arise in communities where one or both of these conditions is not the case. The composition then becomes a vehicle by which the administrator may define traffic "problem" situations in the environment.

The manner in which compositions are used to specify the complete time and spacial environment is discussed below in Composition to Region Assignment.

Just as the hour is the basic time unit, the "region" is the fundamental unit in the environmental specification. That is, composition environmental parameters are assumed constant over both regions of the city and hours of the day.

A region is a subset of the city area. Like the city limit definition, the boundaries must be constructed from sides of a coordinate rectangle. Regions may take the form of a single intersection, lengths of streets, or areas in general. They are defined to achieve the desired spacial distribution of parameters over the city. Each region is identified by a unique integer and constructed from coordinate rectangles. Examples of regions include:

1. residential areas
2. commercial areas
3. main arterials
4. trouble intersections

These regions do not take on "life" until the compositions containing their environmental characteristics have been assigned as described below in the next section.

Composition to region assignments for each daily pattern

The assignment of composition to regions for each daily pattern is the linkage which ties together the previous elements and completes the description of the urban traffic system in both the time and space dimensions. It is here that the advantage of using the artificial device of the composition to reduce the amount of information input is fully realized. The same composition can be sprinkled throughout the city at various hours of the daily patterns.

Composition assignment is probably best understood through an example. Suppose region 1 consists of residential areas. Composition 1 represents offpeak hours in these areas, with a low intersection flow rate and few types of violations. Composition 2 represents peak hours and has higher intersection flow rates and more types of violations, for example, speeding on the way home from work. During daily pattern weekday except for rush hours Composition 1 might apply, Composition 2, during the rush hours. Clearly, in a commercial area Composition 1 might never apply, or perhaps only apply in the wee hours of the morning.

The administrator can, thus, use the six tools discussed with which to create a seemingly real urban traffic environment.

6.2 Player Input

The objective of the players in the game situation is to allocate their available manpower through beat definitions which "solve" the problems the administrator has built into the environment. The player varies unit assignments and concentrations with each game play in an effort to achieve an "optimal" system, in terms of time and space distribution, patrol beat size, etc. Manpower must be optimally allocated to cover routine traffic control problems or normally congested streets or intersections, as well as accident coverage and normal or emergency patrol activity within each traffic area. Given an initial report and reports generated after each play, the team must establish policies and allocate given, limited manpower during each play. The players are, thus, forced to absorb large amounts of data, gain an overall perspective, and acquire a sense of interrelationship between the total system and its parts. The following elements make up the decision input by players at each step in the discontinuous play.

1. district definition
2. area definition
3. beat assignment

A discussion of each of these follows:

District definition

After the players have been presented with the urban traffic environment, they must organize themselves into a traffic department. While organizing their departments, the players have the option of dividing the city into districts.

A district is defined in a manner identical to that used by the administrator to define regions. That is, it is a subset of the city area constructed from sides of coordinate rectangles. At this point, the district headquarters' location can also be inputed by the players.

Area definition

The game model assumes three eight hour shifts as follows:

1. 12 midnight to 8 a.m.;
2. 8 a.m. to 4 p.m.;
3. 4 p.m. to 12 midnight.

Each shift commander in each district must designate areas within the district he thinks are important for the assignment of patrol units. One or more patrol units will be assigned to each area for at least one hour during the shift. The collection of area assignments for one unit during the shift is the "beat" for that unit. It is assumed that these areas will usually vary with each game play as the district commanders strive for a "best" area designation.

Areas are defined by district commanders for all shifts within each daily pattern. Areas are limited to coordinate rectangles for simplicity. This limitation does not appear severely restrictive, as the commander has the flexibility to change a unit's area assignment hourly, as discussed below. Coordinate rectangles can be selected for spot (one intersection), line (a length of street), or area patrol.

Unit beat assignments

A unit beat is a time ordered collection of areas assigned to the unit during the shift period. The district shift commander has the ability to assign units to different areas for each hour of the shift. A unit may be assigned to a different rectangle for each hour, to the same rectangle for the entire shift, or any combination thereof. Beats, thus, have a time and a space dimension.

It is expected that the player commander will vary unit beat assignments with each game in an effort to achieve an "optimal" system operation.

It is believed that these three facilities, just discussed, provide most of the controls exercised by a commander in a real life situation. It is worth emphasizing once again, that the game model has been constructed to provide a definite response to controls, so that the student can derive meaning from playing the game. It is hoped that this response has the appearance of reality, but the model itself does not represent a real life environment. Thus, model responses should not be construed as those which may occur in a real life situation.

7. The Output

Using example traffic reports from Fisher and Mosher [6] and the Traffic Institute, the following format has been created for the output of the game. This output presents feedback in response to the decisions made by the players on effective police allocation throughout their reference city.

The report generated is called an Activity Summary. One Activity Summary Sheet is provided for each shift on each day of each daily pattern type. Suppose the game is being played for a simulated month, then, for example, if there are four Sundays in this simulated month, an Activity Summary is generated for Sunday number 1, Shift 1, 2, 3, Sunday number 2, Shift 1, 2, 3 ..., Sunday number 4, Shift 1, 2, 3. The output on the Summary includes: Time, located, violation, accident, response time, service time, and unit identification, which are discussed below.

Time

Time of the violation which was detected or resulted in an accident. Time is the hour of the day where:

1 a.m. is 1
2 a.m. is 2
.
.
12 noon is 12.
1 p.m. is 13
.
.
.
12 midnight is 24

The fractional part of the hour, i.e., five minutes after, etc., is indicated by a number from 1 to 59. Thus, if the first entry in the time column is 2-54, this would indicate the first accident or detected violation occurred at 2:54 a.m.

Location

Location simply indicates the avenue and street intersection where the detected violation or accident occurred.

Violation Identification

Violation ID is the identification number assigned to the violation by the administrators.

Accident

A "Yes" entry indicates an accident resulted from this violation. A "No" entry indicates a detected non-accident producing violation.

Response Time

Response time is the time required for the closest (in time) patrol unit to move from his present location to the scene of the accident, in minutes. A "0000." entry appears when a detected violation has occurred, since by the structure of the simulation, violations are detected only when the unit and the violation are simultaneously at the same intersection, i.e., instantaneous response time.

Service Time

Service time is the time required in minutes to service the violation or accident. Detected violations receive tickets and may involve intoxication tests, short lectures, or radioing and waiting for car and license identification. Accidents may involve first aid, calls for ambulance, call for towing, interviews with witnesses, and so on, including issuing tickets.

Unit Identification

Unit ID is the identification assigned by the player to the patrol unit, indicating to the shift and district commanders what unit covered the violation or detection.

Thus, the information contained in this

Activity Summary provides the players a basis for analyzing the effectiveness of their past decisions and directing their next decisions.

8. Summary

In summary the logical flow of the game is as follows:

1. Administrators make decisions on the parameters for the reference city
2. Players participate in a briefing Players make their decisions on organization, planning, manpower allocation, areas, and beat assignments.
3. Administrator and player inputs are keypunched onto data processing cards.
4. Deck submitted.
5. Model from section 5 generates feedback in the form of Activity Summary Sheets.
6. Certain administrator and player input stored on tape for use in next play of game
7. Reports returned to players, who base new decisions on them
8. New player decision made and deck resubmitted, etc.

The logical flow assumes the city is the same, unless the administrators call a new briefing for a new city.

Notice that the administrators always have the option to change certain parameters when they want to, i.e., increase intersection flow rates on side streets on the day of a parade.

9. Some Final Comments

The Traffic Police Management Training Game, as presented in the preceeding sections of this paper, was played as an experimental effort at the Institute during the final 3 weeks of a year's program. The general reaction from the Traffic Institute staff was extremely favorable.

The administrators are sold on the use of a management game carefully integrated into their curriculum in the future. Since, as far as can be determined, the Traffic Police Management Training Game is unique, the administrators can foresee many possibilities of its use by major Police Departments, state and city, across the country. Further, they can see numerous applications and adaptations.

The player reaction was equally enthusiastic with the positive reservation that they felt more time should be devoted to the game, so that it becomes an integral mechanism for tying the total program together.

The future of the Traffic Police Management Training Game, thus, depends on the successful alteration of the Institute's program to incorporate the game as a teaching tool.

BIBLIOGRAPHY

Books

1. Broom, Halsey, Business Policy and Stratetic Action. Englewood Cliffs, N.J.: Prentice-Hall, 1969.

2. Cohen, Kalman, et al. The Carnegie Tech Management Game. Homewood, Ill.: Richard D. Irwin, Inc., 1964.

3. Dale, A.G. and Klasson, C.R., Business Gaming: A Survey of American Collegiate Schools of Business. Bureau of Business Research University of Texas, 1964.

3. Dill, William R., et al. Proceedings of the Conference on Business Games. Ford Foundation and Tulane University, 1961.

5. Duke, Richard D., Gaming-Simulation in Urban Research. Public Institute of Community Development and Services, Michigan State University, 1964.

6. Fisher, G.R. and Mosher, W.W., Statistical Analysis of Accident Data as a Basis for Planning Selective Enforcement -- Phase II. ITTE Research Report 51, 1964.

7. Frost, Thomas, A Forward Look in Police Education. Springfield, Illinois: Charles C. Thomas, 1959.

8. Greenlaw, Paul; Herron, Lowell; and Rawdon, Richard, Business Simulation in Industrial and University Education. Englewood Cliffs, N.J.: Prentice-Hall, 1962.

9. Guetzkow, Harold, et al., Simulation in International Relations: Developments for Research and Teaching. Englewood Cliffs, N.J.: Prentice-Hall, 1963.

10. Guetzkow, Harold and Cherryholmes, Cleo, Inter-Nation Simulation Kit. Chicago: Science Research Associates, 1966.

11. Guetzkow, Harold, Simulation in Social Science. Englewood Cliffs, N.J.: Prentice-Hall, 1962.

12. Harrison, Leonard, How to Teach Police Subjects: Theory and Practice. Springfield, Ill.: Charles C. Thomas, 1964.

13. Henshaw, Richard and Jackson, James, The Executive Game. Homewood, Ill.: Richard D. Irwin, Inc., 1966.

14. Kaplan, Morton A. (ed.), New Approaches to International Relations. New York: St Martin's Press, 1968, p. 202-269: "Some Correspondences between Simulations and "Realities: in International Relations", Harold Guetzkow.

15. Kibbee, Joel; Craft, Clifford; and Nanus, Burt, Management Games -- A New Technique for Executive Development. New York: Reinhold Publishing Corp., 1961.

16. Klotter, John, Techniques for Police Instructors. Springfield, Ill.: Charles C. Thomas, 1963.

17. Mulvihill, Donald (ed.), Guide to the Quantitative Age. New York: Hold, Rinehart, and Winston, Inc., 1966.

18. North, Robert, et al. Content Analysis: A Handbook with Application for the Study of International Crisis. Evanston, Ill.: Northwestern University Press, 1963.

19. Raser, John R., Simulation and Society. Boston: Allyn and Bacon, Inc., 1969.

20. Schellenberger, Robert Earl, Development of a Computerized Multipurpose Retail Management Game. Research Paper 14. University of North Carolina at Chapel Hill, Feb., 1965.

21. Smith, W. Nye; Estey, Elmer E.; and Wines, Ellworth F., Integrated Simulation. Cincinnati, Ohio: South-Western Publishing Co., 1968.

22. Thorelli, Hans and Graves, R., International Operations Simulation. New York: The Free Press of Glencoe, 1964.

23. Vance, Stanley, Management Decision Simulation -- Noncomputer. New York: McGraw-Hill, 1960.

Articles and Periodicals

24. Anglinger, G.R., "Business Games--Play One". Harvard Business Review, 36 (March-April, 1958), 115-125.

25. Anglinger, G.R., and Vollman, Thomas, "Uniproduct: A Pedagogical Device". California Management Review, 10 (Winter, 1967), 65-70.

26. Babb, E.M.; Leslie, M.A.; and Van Slyke, M.D., "The Potential of Business-Gaming Methods in Research". Journal of Business, (October, 1966), 465-472.

27. Bankhead, K. and Herms, B.F., "Reducing Accidents by Selective Enforcement". Traffic Digest and Review, 18 (January, 1970), 1-5.

28. Barish, N. and Siff, Frederick, "Operational Gaming Simulation with Application to the Stock Market". Management Science, 15 (July, 1969), 530-541.

29. Boguslaw, Robert; Davis, Robert and Glick, Edward, "A Simulation Vehicle for Studying National Policy Formation in a Less Armed World". Behavioral Science, 2 (January, 1966), 43-61.

30. Carlson, Elliot, "Versatile Business Gaming". Management Review, 55 (September, 1966), 45-47.

31. Carson, John, "Business Games: A Technique for Teaching Decision-Making". Management Accounting, 49 (October 1967), 31-35.

32. Cherryholmes, Cleo, "Some Current Research on Effectiveness of Education Simulations: Implications for Alternative Strategies". American Behavioral Scientist, 10

(October, 1966), 4-5.

33. Churchill, Neil and Cyert, Richard, "Experiment in Management Auditing". *Journal of Accounting*, 121 (Fall, 1966), 39-43.

34. Cohen, K., *et al*. "The Carnegie Tech Management Game". *Journal of Business*, 33 (October, 1960), 303-309.

35. Collett, Merrill, "Simulation As a Management Development Tool". *Personnel Administration*, 25 (March, 1962), 48-51.

36. Coplin, William, "Inter-Nation Simulation and Contemporary Theories of International Relations". *American Political Science Review*, 60 (September, 1966), 562-578.

37. Crow, Wayman J., "A Study of Strategic Doctrines Using the Inter-National Simulation". *The Journal of Conflict Resolution*, 7 (September, 1963), 580-589

38. Dolbear, F.T.; Attiyeh, R.; and Brainard, W.C., "A Simulation Policy Game for Teaching Microeconomics". *American Economic* Review, 58 (May, 1968), 483-491.

39. Druckman, Daniel, "Ethnocentrism in the Inter-Nation Simulation". Evanston, Ill.: Simulated International Process Project, Northwestern University, 1967; *The Journal of Conflict* Resolution, 12 (March, 1968).

40. Eilon, Samuel, "Management Games". *Operations Research Quarterly*, 14 (June 1963) 137-149.

41. "Game Teaches Salesmen". *Sales Management*, 101 (September, 1968) 139-140.

42. Guetzkow, Harold, *et al*. "An Experiment on the N-Country Problem Through Inter-Nation Simulation". St. Louis, Missouri: Washington University, 1960.

43. "HOCUS: The Management Parlour Game". *Business Management*, 99 (April, 1969), 24-27.

44. Hodgetts, Richard, "Management Gaming for Didactic Purposes". *Simulation and Games*, 1 (March, 1970).

45. Klasson, Charles, "Business Gaming: A Progress Report". *Academy of Management Journal*, (September, 1964), 175-188.

46. Lewin, A. and Weber, W., "Management Game Teams in Education and Organization Research: An Experiment on Risk Taking". *The Management Sciences Research Group*, Carnegie-Mellon University, Pittsburgh, Pennsylvania, November, 1968.

47. "Management Game Puts Team on Firing Line". *Administrative Management*, 29 (August, 1968), 31.

48. McKenney, James, "Evaluation of a Business Game in an MBA Curriculum". *Journal of Business*, 35 (July, 1962), 278-286.

49. Moffie, D. and Levine, R., "Experimental Evaluation of a Computerized Management

Game". *The Atlanta Economic Review*, 18 (November, 1968).

50. "Profits Set Score at Business School Tournament". *Business Week*, (March 18, 1967), 156-158.

51. Raia, Anthony, "Study of the Educational Value of Management Games". *Journal of Business*, 39 (July, 1966), 339-352.

52. Rausch, Erwin, "Games Managers Play". *Administrative Management*, 29 (December, 1968), 36.

53. Shubik, Martin, "Gaming: Costs and Facilities". *Management Science*, 14 (July, 1968), 629-660.

54. Stanley, John, "Management Games: Education or Entertainment?" *Personnel Journal*, 41 (January, 1962), 15-17.

55. Symonds, Gifford, "Study of Consumer Behavior by Use of Competitive Business Games". *Management Science*, 14 (March, 1968), 473-485.

56. Taxel, Hal. "Let's Play Games". *Editors and Publishers*, 101 (March, 1968), 53.

57. Thorelli, Hans; Graves, Robert; and Howells, Lloyd, "INTOP". *Journal of Business*, 35 (July, 1962), 287-297.

58. "Traffic Police Administrator Training Program". *Traffic Digest and Review*, 18 (February, 1970), 16.

59. Tuason, Roman, "MARKAD" A Simulation Approach to Advertising". *Journal of Advertising Research*, 9 (March, 1969), 53-58.

60. Vance, Stanley and Gray, Clifford, "Use of a Performance Evaluation Model for Research in Business Gaming". *Academy of Management Journal*, 10 (March, 1967), 27-37.

61 Van Dyck, J., "Understanding the Organizational Process Via the Management Game". *Journal of Management Studies*, (Oxford, England), 5 (October, 1968), 338-351.

Theses and Dissertations

62. Chadwick, Richard W., "Developments in a Partial Theory of International Behavior: A Test and Extension of Inter-Nation Simulation Theory". Ph.D. Dissertation. Evanston, Illinois: Department of Political Science, Northwestern University, June, 1966.

63. Hermann, Margaret, "Stress, Self-Esteem, and Defensiveness in an Inter-Nation Simulation." Ph.D. Dissertation. Evanston, Illinois: Department of Psychology: Northwestern University, 1965.

64. Sherman, Allen William. "The Social Psychology of Bilateral Negotiations". Master of Arts Thesis. Evanston, Illinois: Department of Sociology, Northwestern University, 1963.

Pamphlets

65. Inter-Nation Simulation Participant's

Manual, International Relations Program, Department of Political Science, Northwestern University.

66. "Traffic Police Administration Training Program". Distributed by The Traffic Institute of Northwestern University, 1970.

67. "1968-1969 Training Calendar". Distributed by The Traffic Institute of Northwestern University.

68. "The UCLA Executive Decision Game Participant Game Information."

Unpublished Materials

69. Chadwick, Richard, "Relating Inter-Nation Simulation Theory with Verbal Theory in International Relations at Three Levels of Analysis". Evanston, Illinois: Simulated International Processes project, Northwestern University, July, 1966.

70. Chadwick, Richard, "Extending Inter-Nation Simulation Theory: An Analysis of Intra- and International Processes project, Northwestern University, August, 1966.

71. Chadwick, Richard, "Theory Development Through Simulation: A Comparison and Analysis of Associations Among Variables in an International System and an Inter-Nation Simulation". Evanston, Illinois: Simulated International Process project, Northwestern University, September, 1966.

72. Croke, E., et al., The Use of Gaming Techniques in the Traffic Police Administration Training Program at the Northwestern University Traffic Institute. IE/MS D30-1, Class Project (Fall, 1968).

73. Elder, Charles and Pendley, Robert, "Simulation as Theory Building in the Study of International Relations". Evanston, Illinois: Simulated International Processes project, Northwestern University, July, 1966.

74. Elder, Charles and Pendley, Robert, "An Analysis of Consumption Standards and Validation Satisfaction in the Inter-Nation Simulation in Terms of Contemporary Economic Theory and Data". Evanston, Illinois: Simulated International Processes project, Northwestern University, November, 1966.

75. Gorden, Morton, "International Relations Theory in the TEMPER Simulation". Evanston, Illinois: Simulated International Relations project, Northwestern University, 1967.

76. Kennedy, Allen S., Specifications for a Traffic Police Management Training Game. IE/MS D99, Independent Project (Spring, 1969).

77. Kress, Paul, "On Validating Simulation: With Special Attention to the Simulation of International Politics". Evanston, Illinois: Northwestern University, 1965.

78. Meier, Dorothy, "Progress Report: Event Simulation Project". Evanston, Illinois: Simulated International Process project, Northwestern University, 1965.

79. Nardin, Terry and Cutler, Neal, "A Seven Variable Study of the Reliability and Validation of Some Patterns of International Interaction in the Inter-Nation Simulation". Evanston, Illinois: Simulated International Process project. Northwestern University, December, 1967.

80. Pendley, Robert and Elder, Charles, "An Analysis of Office Holding in the Inter-Nation Simulation in Terms of Contemporary Political Theory and Data on the Stability of Regimes and Governments". Evanston, Illinois: Simulated International Process project, Northwestern University.

81. Smoker, Paul, "An International Process Simulation: Theory and Description". Evanston, Illinois: Simulated International Processes project: Northwestern University, 1968.

SIMULATION: METHODOLOGY FOR THE SYSTEM SCIENCES

G. Arthur Mihram, Ph.D.

Faculty of
The University of Pennsylvania
Philadelphia 19104

and

Consultant to
Operations Research, Incorporated
Silver Spring, Maryland 20910

Sessions:

α. Systemic Analysis and Model Synthesis
β. Simulation: Ensuring its credibility
γ. System Scientists: Executives in disguise

0. LEXICON

A. Simulation (dynamic, stochastic): A generator of realisations from a time series, or stochastic process, of generally unknown characteristics.

B. Credibility: That quality of a literal model which is extablished as a result of:
(1) its concise, lucid, logical, and unambiguous expression; and,
(2) its precise reproduction of the simuland's observable behaviour.

C. System: A collection of interdependant and interactive elements which together behave in a collective effort to attain some (usually specifiable) goal.

D. System Science: The search for explanations for (i.e., credible models of) systemic behaviour.

E. Simular: Of, or having to do with, a simulation model. In contrast with:

F. Systemic: Of, or having to do with, the system (or simuland).

G. The Scientific Method: The operational procedure by which man augments his knowledge of the World:

I. Systems analysis - The isolation of the salient components, interactions, relationships, and behaviour mechanisms of a system;

II. System synthesis - The grammatical and logical organisation of the literal representation of a system's behaviour, in accordance with the findings of the preceding systems analysis stage;

III. Verification - the comparison of the model's responses with those anticipated if indeed the model's structure were programmed as intended;

IV. Validation - the comparison of recorded observations of the simuland with the simular responses emanating from independently seeded encounters of the verified model, thereby establishing the verisimilitude of the model and the modelled;

V. Inference - the fifth (and most rewarding) stage of a model's development, concerned with designed, simular experimentation, employing independently seeded encounters with the verified and validated model.

Ω. Null stage - At the outset of a scientific enquiry, a statement of modelling goals, acknowledging the system scientist's accumulated neural recording of his religious, societal, and cultural upbringing, plus that of his personal experience with nature and that of his exposure to precedent scientists via educational institutions and their libraries.

α. SYSTEMIC ANALYSIS AND MODEL SYNTHESIS

I. SYSTEMS ANALYSIS

- A. Definition of systemic goal(s).
- B. Delineation of primary systemic attribute(s).
- C. Study of primary systemic event(s).
- D. Specification of secondary systemic attributes.
- E. Determination of secondary systemic events.
- F. Et Cetera: The Uncertainty Principle of Modelling.
- G. Circumscription of boundary for systemic feedback mechanisms.

A Note Aside: The Cornerstone of Successful Modelling is the set of instructions for the collection of data that shall be required by the programmed (written) model.

II. SYSTEM SYNTHESIS

- A. The Simulation's Executive Algorithm. [Cf: Figure 2.].
- B. Definition of state variables.
- C. Formulation of operationally defined event algorithms (exogenous and endogenous):
    - 1. Test section;
    - 2. Action section;
    - 3. Report section.
- D. Stochasticity - seed phyla:
    - 1. Histograms;
    - 2. PDF's;
    - 3. Series records;
    - 4. Stochastic process.
- E. Specification of initial conditions.
- F. Selection of an ad hoc simulation programming language.

A Note Aside: Model completion can be considerably expedited by means of contemporaneous programming and data collection efforts. [Cf: Figure 2.].

β. SIMULATION: ENSURING ITS CREDIBILITY

III. MODEL VERIFICATION

- A. Grammatical rectification.
    - 1. Orthography: at the keypunch machine.
    - 2. Syntactic analysis.
- B. Logical veracity.
    - 1. Ambiguity exclusion: the language faults corrected.
    - 2. Deterministic verification tests.
- C. Fundamental Principium of Seeding
    - 1. Random samples.
    - 2. One-sample statistical tests.
- D. The Cycling Between Modelling Stages. [Cf: Figure 1.].

IV. MODEL VALIDATION

- A. The Simuland Sampled: The Uncertainty Principle of Modelling Revisited.
- B. The General Principium of Seeding:
    - 1. Experimental error.
    - 2. Two-sample statistical tests.
- C. The Cycling Among Modelling Stages. [Cf: Figure 1.].

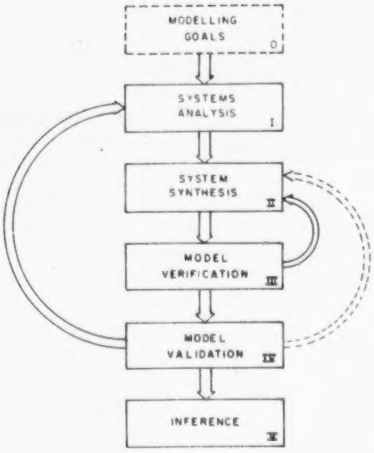


Figure 1. SCIENTIFIC METHOD
(or: CYBERNETICS OF EPISTEMOLOGY)

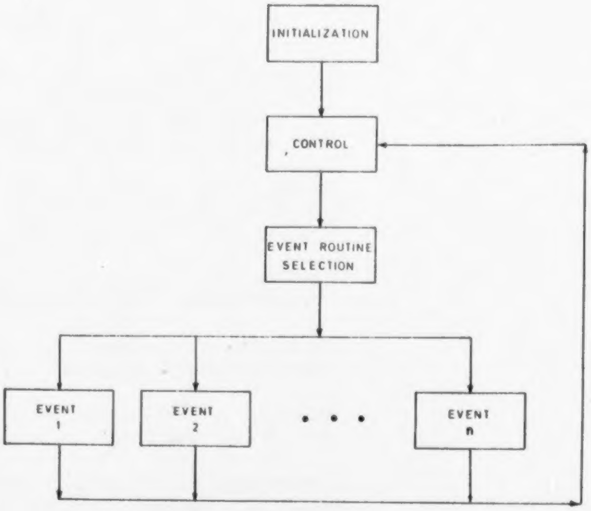


Figure 2. THE SIMULATION ALGORITHM
(or: CYBERNETICS OF EFFECTIVE EXECUTIVES)

Y. SYSTEM SCIENTISTS: EXECUTIVES IN DISGUISE

V. INFERENCE

- A. Modelling Goals: The Null Stage Revisited.
- B. Analysis of Static Effects
  - 1. Point and interval estimation of distributional properties.
  - 2. Forecasting the distribution of systemic yields.
  - 3. Prediction of the significance of differences in systemic operating conditions: Analysis of variance, regression, multivariate analyses.
  - 4. Foretelling preferabilities among operating conditions: Multiple ranking and comparison techniques.
  - 5. Prophesying optimal operating conditions: Response surface methodology.
- C. Analysis of Dynamic Effects
  - 1. Trend predictions: their disclosure and removal.
  - 2. Systematic fluctuations: spectral analysis.
- D. The Augmentation of Human Knowledge [Cf: Figure 1.].

MANAGEMENT IN SCIENTIFIC PERSPECTIVE

- A. Planning: The Scientific Method [Cf: Figure 1.]
- B. Programming: The Executive Algorithm [Cf: Figure 2.]
- C. Budgeting: The Regrettable, but Necessary, Constraint.

REFERENCES

Mihram, G. Arthur (1972), SIMULATION: STATISTICAL FOUNDATIONS AND METHODOLOGY, volume 92 of Professor Richard Bellman's series, Mathematics in Science and Engineering, Academic Press, New York.

Maisel, H. T. and G. Gnugnoli (1972), SIMULATION OF DISCRETE STOCHASTIC SYSTEMS, Science Research Associates, Palo Alto, California.

Dutton, J. M. and W. H. Starbuck (1971), COMPUTER SIMULATION OF VOTING BEHAVIOR, John Wiley, New York.

Naylor, T. H., editor (1971), COMPUTER SIMULATION EXPERIMENTS WITH ECONOMIC SYSTEMS, John Wiley, N. Y.

Reitman, J. (1971), COMPUTER SIMULATION APPLICATIONS, John Wiley, New York.

Stephenson, R. E. (1971), COMPUTER SIMULATION FOR ENGINEERS, Harcourt, Brace, Jovanovich, New York.

Emshoff, J. R. and R. L. Sisson (1970), DESIGN AND USE OF COMPUTER SIMULATION MODELS, Macmillan, N. Y.

Mihram, G. Arthur (1972), "Some Practical Aspects of the Verification and Validation of Simulation Models", OPERATIONAL RESEARCH QUARTERLY, 23, 17-29.

Evening 1: Night in a Berkeley Laboratory
Chairman: Austin Curwood Haggatt

A novel micro-programmed APL computer for laboratory control is being developed in the Laboratory under sponsorship of the National Science Foundation. Visitors will have an opportunity to see a demonstration of the system and hear a presentation of its design features.

Paper: "A Micro-Programmed APL Language Laboratory Control System"
Austin Haggatt, Mark Greenberg, University of California
Jeffrey Moore, Stanford University

In addition there will be two presentations on joint teaching and research use of business games.

Papers
"A Business Game for Teaching and Research: Some Experiences"
Martin Shubik, Yale University

"The New York University Business Game"
Myron Uretsky, New York University

Evening 1 Night in a Berkeley Laboratory

Chairman Austin Curwood Hoggatt, University of California

In this session we examine the impact of two large management games on the teaching and research function of two large teaching institutions, New York University and Yale University. We conclude with an examination of the development of an APL language processor which is to be the control facility for an already established man-machine laboratory for the study of human decision making in complex environments.

A Business Game for Teaching and Research: Some Experiences

Martin Shubik, Yale University

This paper discusses an approach to gaming experimentation and teaching based upon the utilization of one specific type of game and the construction of a time sharing system designed to provide students and experimenters ease in playing the game, modifying the game and analyzing the output.

A Micro Programmed APL Processor for Experimental Control of Man-Machine Interaction

Austin Hoggatt, Mark Greenberg, Center for Research in Management Science, University of California
and Jeffrey Moore, Stanford University

The needs for laboratory control of man-machine interaction form a super set of the needs which lead to time sharing systems. In addition to facilities for on line construction and debugging of programs and operating systems aids to the allocation of resources, there are real time constraints on system response in order that subjects are placed in environments which facilitate the recording of observations on their behavior with minimal delays or machine induced artificiality of situations. Requirements for console interaction and control of physical facilities (tape recorders, t.v. monitors, special input output devices) require the development of an operating system which accomodates these needs. A need to describe models of the subject environment in an operational publication language led to the adoption of APL as the experimenter's language to which have been added extensions to allow parallel operation of instances of the same process, multiple I/O units under control of a single process, and control to provide for protection of the integrity of ongoing experiments. Reasons of economic viability dictated that this be accomplished on a low budget system. Under N.S.F. funding the CRMS Laboratory is pursuing the development of a pair of micro-programmed systems on which respectively are implemented the operating system and the APL engine. We emphasize the interplay between needs of experimenters, software and hardware design and the allocation of functions between hardware and software in order to minimize the difficulties of realization in the resulting system.

EXPERIENCES IN TRANSFERRING A MANAGEMENT GAME ACROSS CULTURAL AND ECONOMIC BOUNDARIES

Myron Uretsky, New York University

The New York University Management Game is a complex man machine simulation. It is based upon the original Carnegie Tech Game, but has since been extensively modified. One of the most significant aspects of this particular simulation is its extensive interaction with the surrounding business community: Boards of Directors are utilized, consisting of high level business men; financing is handled through live negotiations with major banks, insurance companies, and underwriters; federal regulatory agencies monitor and investigate Game-related activities. Legal counsel is available together with the provision of judicial services. Auditing and labor union activities are provided through other courses being offered.

The heavy involvement of external participants has made it possible to modify this Game in order to meet the needs of Polish and Hungarian educational institutions. The programming modifications have been accomplished in a straight-forward manner. Involvement of Polish and Hungarian businessmen and ministry officials has provided automatic adjustments for some socio-economic factors. At the present time a working version is operational in Budapest and a Polish version is expected to be tested in late January. A number of behaviorally oriented research projects are now being carried out in parallel in the United States and in eastern Europe. These studies represent attempts to determine the impact of Management Gaming on eastern European businessmen, and an attempt to isolate differences in business behavior in these two societies.

Session 13: Management Applications

Chairman: Arnold Ockene, Securities Industries Automation Corporation

This session presents four papers dealing with models for use by upper management. The first is an evaluation of simulation as a tool for corporate financial planning, and includes a discussion of optimization models. This is followed by a description of a general university simulation program for the study of resource allocation and utilization problems. The final two papers are concerned with gas utilities. One deals with the relationships between rate, revenue, and cost of service; the other with the investment problems facing gas pipeline companies.

Papers

"Corporate Simulation Models - A Reappraisal"
Willaim F. Hamilton, University of Pennsylvania

"A General University Simulation Model"
Charles C. Daniel, National Aeronautics and Space Administration
Hamid K. Eldin, Oklahoma State University

"Rate-Revenue-Cost of Service Simulation of a Natural Gas Utility"
D. Jeffrey Blumenthal, On-Line Decisions, Inc.

"A Simulated Investment Analysis for a Gas Pipeline Company"
H. J. Miller, Colorado Interstate Gas Company

Discussants

John Lastavica, First National Bank of Boston
Stanley G. Reed, IBM Corporation
Harold Steinberg, IBM Corporation
Henry Lande, IBM Corporation

CORPORATE SIMULATION MODELS - A REVIEW AND REAPPRAISAL

William F. Hamilton

Department of Management
The Wharton School
University of Pennsylvania
Philadelphia, Pennsylvania 19104

ABSTRACT

This paper examines the potential and the practice of simulation modeling for corporate financial planning. The contributions of formal models to the planning process are reviewed and appropriate design features are proposed. Comparative analysis of simulation and optimization applications in corporate planning points to substantial functional complementarity and opportunities for combined use. An actual planning system involving both simulation and optimization techniques is described to illustrate the feasibility and potential of the hybrid modeling approach.

Computer-based corporate models have generated considerable interest among management scientists and corporate planners in recent years [9,20]. Not surprisingly, the size and complexity of corporate-level planning problems have favored the development of simulation models for evaluating the implications of selected planning alternatives. Recent studies indicate that such models permit planners to consider a far greater number of alternatives in detail and with greater confidence than is possible using traditional planning methods [10,18].

The potential benefits of corporate modeling are impressive, but a review of current applications reveals a substantial gap between the potential and the practice. In many cases, models have been judged useful in improving corporate planning and worth the costs of development and application. At the same time, there is mounting evidence that some modeling efforts have fallen far short of their potential contributions [11,12]. The reasons are many and varied, ranging from technical model design to organizational deficiencies.

The role of formal models in corporate planning and basic model design considerations are reviewed in the next section. This is followed by a discussion of the relative merits of simulation and optimization methods and their functional complementarity in corporate modeling. The paper ends with a description of a corporate model system which incorporates both simulation and optimization capabilities. This system has demonstrated that effective analytical support of the planning process can be achieved through combined use of modeling and information system technologies.

MODELS IN CORPORATE PLANNING

In many companies, support for the development of a corporate model has generally grown out of frustration with the inherent shortcomings of manual plans preparation [4]. Corporate planning is often limited in manual systems to simple analysis of a few relevant variables and assumptions. Both time and resource requirements typically preclude detailed consideration of more than a few planning alternatives. Continuous revision of established plans and evaluation of new opportunities in changing competitive situations is virtually impossible. Too often the information available for planning decisions is inadequate and irrelevant, placing undue reliance on judgement and intuition. As a result, there has been increasing recognition of the need to improve both the planning process and its operation.

Uses of Corporate Planning Models

Corporate models are constructed and applied in order to make inferences about future performance of the corporate system. In contrast to the system itself, a model can be manipulated easily by modifying inputs and other parameters describing the system and its planning environment to allow estimation of the impact of such modifications.

Corporate planning requires the identification, evaluation, and selection of alternative courses of action. In order to plan effectively, corporate management must understand the interactions between corporate activities and the effects of decisions on these activities and on overall corporate performance. They must examine the many alternative courses of action which are available, and they must anticipate and be able to respond quickly to changing conditions. In this context, a computer-based corporate model can serve a number of useful purposes. Those most frequently cited by corporate model builders and users include:

(1) rapid and accurate evaluation of planning alternatives;

(2) prediction of the effects of changing environmental conditions;

(3) estimation of the sensitivity of corporate performance to planning assumptions;

(4) screening and evaluation of acquisition candidates;

(5) development of insights into the complexity of corporate activities and interactions.

These uses of a corporate model reflect its primary role as a tool to assist in the execution of the planning process. By reducing the time and resource requirements for plan evaluation, a model can facilitate the consideration of a number of alternatives rather than just a limited few. This can be an important step toward improving the quality of corporate planning.

The development and implementation of a corporate planning model can also have important implications for improving the planning process itself. As Ackoff has noted [1]:

> The principal contribution of scientists to planning may not lie in the development and use of relevant techniques, but rather in their systemization and organization of the planning process, and in the increased awareness and evaluation of this process that their presence produces.

A model can contribute significantly toward this end. It provides a systematic and explicit structure to the planning process and requires corporate planners to specify and coordinate their planning assumptions, logic, and data requirements. It can thus help to ensure internal consistency and reproducibility among the plans of diverse corporate groups. A model also formally represents the current state of knowledge about the corporation and facilitates communication in the planning process by providing a common framework for discussion and analysis.

It is essential to recognize, however, that many important planning functions lie beyond the scope of even the most sophisticated formal models. Most models provide increased power to explore and evaluate planning alternatives, but the identification, selection, and implementation of alternatives remain critical management responsibilities. The corporate plan can be no better than the set of alternatives considered, and the outcome of the planning process depends ultimately on successful implementation of the selected alternatives. Similarly, the development and evaluation of model inputs and assignment of appropriate values to judgemental factors are required before meaningful model studies can be conducted. A very real danger associated with the development and use of a corporate model is the tendency toward overemphasis on the computational aspects of the planning process at the expense of more fundamental and important considerations requiring management judgement and intuition. If properly used, however, a corporate model can enhance and encourage, rather than limit, creative management inputs to planning.

Model Characteristics

At least two important issues arise in attempting to develop a corporate model capability consistent with the uses indicated above. These relate to the intended "user" of the model and its relationship to existing manual planning practices.

Corporate planning models are typically designed for one of two potential users: executives with ultimate responsibility for planning decisions, or staff planners. Most modeling efforts to date have been directed toward developing a tool for staff planners [4, 10], rather than one which is appropriate for use by managers [3, 5]. Although involved in the same planning process, these two groups have different requirements and are likely to use models differently. For example, the planner is more likely to be interested in detailed analyses and specialized outputs which reveal the nature of subsystem interactions, while the manager will generally prefer aggregated outputs in familiar formats. The manager is also less likely to understand detailed or sophisticated models which might be more appropriate for the planning studies conducted by his specialized planning staff. It is therefore essential that the intended user(s) be clearly identified at an early stage in model design. Current opinion seems divided as to the most appropriate user, and there is an apparent trend toward systems which permit partitioning of operating requirements and outputs to meet the needs of both potential user groups.

Opinion is also divided on the extent to which a planning model should automate existing planning procedures without imposing additional requirements or changes. Computer-based models can be designed to permit far more sophisticated analysis of a broader range of planning problems and data inputs than is possible in manual planning systems. Also, models can often make possible "top-down," iterative strategic planning in corporations where existing manual procedures could cope only with a decentralized "bottom-up" planning process. As Hertz [15] has observed:

> Business strategic planning is "top-down," long range corporate planning that challenges the basic goals and directions that have guided the enterprise in the past. More technically stated, it should be prospective decision making, done after the systematic evaluation of all reasonable alternative courses of action...

This is often best accomplished through centralized evaluation of planning alternatives facing the corporation. On the other hand, where formal planning procedures are well established, attempts to implement a model which requires altering these procedures - rather than simply automating existing inputs, calculations and outputs - may be doomed to implementation problems and ultimate failure.

The number and variety of corporate models now in use offer the opportunity to compare model characteristics and select those which seem most appropriate for future modeling efforts. Seven basic design characteristics with particular implications for planning effectiveness and efficiency were identified in the process of developing a corporate planning system [14] and are discussed briefly below:

(1) Scope Although heavily oriented toward the financial aspects of planning decisions, most corporate models reflect the full range of corporate activity over multiple time periods. This scope is essential if important subsystem interactions and long-term implications are to be adequately reflected in planning studies.

(2) Structure The structural characteristics of existing corporate models vary widely [7,20]. In some instances, the model is a single construct which incorporates desired features of the system being planned, often at a considerable level of aggregation. This approach facilitates representation of interactions between corporate subsystems and usually offers some economic advantages. However, the magnitude of the effort also presents the danger that the model will become obsolete or will be abandoned before it can be made operational and useful. Another approach is to construct separate models reflecting different corporate activities and/or planning analyses. These can be linked to each other and to a common planning data base to create a corporate model system. Component models can thus be developed over time and applied in the planning process as they become operational. Moreover, this approach generally permits use of a variety of analytical techniques which could not be accommodated in a single model structure.

(3) Realism If it is to be useful as a planning tool, a corporate model must provide a realistic representation of the system being planned. At the same time, a model is by definition an abstraction from reality. The most appropriate level of abstraction will vary with the intended purpose of the model and the desired results. This implies careful selection of a subset of relevant variables and relationships in model design. Inadequate detail may result in a model with limited usefulness for evaluating plans under changing conditions; too much detail may result in excessive data requirements and development costs. Existing planning models range in abstraction from highly aggregated accounting information compilers to detailed models of corporate operations [2]. The majority of these models consider the total corporation using summary variables (usually financial in nature) rather than representing corporate operations in detail [9]. The most common model outputs are pro forma financial statements or, less frequently, aggregate production plans. Only a handful of models reflect, even in a limited fashion, the stochastic nature of corporate activities and performance. Risk and uncertainty are inherent in the planning process, but most corporate models permit only deterministic projections [9,12,20]. As these "first generation" models are accepted and implemented, however, much greater emphasis on stochastic modeling extensions can be expected. This direction will be further encouraged by continuing advances in both computer and modeling technologies.

(4) Flexibility If it is to be of continuing usefulness, a corporate model must be flexible enough to reflect changes in corporate structure (through organization, acquisition, diversification, etc) or expansion in scope without

extensive development effort. Planning is conducted in a rapidly changing corporate environment and provision must also be made for easily updating planning relationships and data. Another important aspect of model flexibility is its applicability to a wide range of planning problems in both the annual planning cycle and interim studies of new opportunities or changing conditions. A modular modeling approach involving a set of linked submodels is most likely to provide such flexibility.

(5) Ease of Use A corporate model has value as a planning tool only to the extent that it is actually used in the planning process. Wherever possible, therefore, the model should be designed so that it can be easily understood by the user(s) and operated with a minimum of inconvenience. This is especially important where the user is a manager with limited time, patience, and analytical background. Only when the user understands the capabilities and limitations of the model and its data base is it likely to be used effectively or, perhaps, used at all. It is not essential, however, that the user fully understand the technical development or internal logic of the model. The ability to interpret model outputs quickly in light of explicitly stated model assumptions and to compare them with other types of planning information is generally sufficient.

A number of model operating features strongly influence its ease of use and therefore deserve design consideration. Provision for automatic or computer-assisted input generation, data base editing and updating, and output report preparation is essential. Some existing corporate models require days (or even weeks) for input preparation and translation of model outputs into desired formats. This not only greatly increases the time and cost associated with model operation, but also seriously limits effective use of the model as a creative planning tool. Corporate planning is inherently an interactive, investigative process in which intermediate results may indicate appropriate directions for futher analysis. Therefore, access to the model and data base via remote terminals (where economically feasible) and provision for multiple input/output options can greatly facilitate model application.

(6) Resource Requirements The time and cost required for model development, updating/modification, and operation are important considerations which depend heavily upon, and often restrict, other design characteristics. For example, the sophistication and flexibility provided in an initial model version may be limited by budgetary restrictions and by pressures to demonstrate the feasibility of the corporate modeling approach in a short time period. In general, the broader the scope, the more modular the structure, the greater the realism, the greater the flexibility, and the easier the model is to use, the more costly

it will be. As is often the case in modeling efforts, however, the initial costs of design and development must be balanced against both the quality of results and the time and manpower required to generate them.

(7) Capabilities The primary purpose of corporate planning studies is to evaluate alternatives and identify those which in some sense best satisfy corporate performance objectives. This implies both the evaluation and selection of planning alternatives. Corporate models typically assist in this process in one of two ways:

(a) by projecting the implications of pre-selected alternatives, or

(b) by selecting the best alternative(s) from the available set.

This is not simply a play on words. Most existing corporate planning models are computer-based financial simulations which are used to test the feasibility and project the effects of proposed alternatives. The Xerox planning model described by Brown [4], for example, computes the financial implications of alternative marketing and production policies under different environmental conditions and generates projected financial statements for each set of inputs. In all but the simplest cases, there are a great many distinct planning alternatives and combinations of alternatives to be considered. This suggests the desirability of optimum-seeking capabilities to assist in the selection, rather than just the evaluation, of alternatives. As has been noted elsewhere, a great deal more activity can be expected in the development of optimization models for corporate planning [6,7].

The design characteristics discussed above are important determinants of the extent to which corporate models are likely to achieve their potential as planning tools. Further consideration of the nature and relative merits of simulation and optimization methods in corporate modeling follows in the next section.

SIMULATION OR OPTIMIZATION ?

Simulation and optimization have generally been viewed as alternatives in the design and development of corporate models. Selection of the most appropriate approach is a significant step in model design which strongly influences both the functional role and technical structure of the model. As indicated above, the essential functional difference is between an alternative tester (simulation) and an alternative selector (optimization).

Corporate Simulation Models

In his survey of corporate modeling, Gershefski [9] reported that the overwhelming majority (95 percent) of corporate models were "computer simulations which utilize case studies to determine the effect of different strategies." Other studies have confirmed this popularity [7,18,20]. There are many reasons for the widespread acceptance and use of computer simulation in corporate modeling.

Simulation techniques are applicable in situations which are too complex for analytical formulations and, in general, permit a greater degree of model realism in other cases. Moreover, the development and application of a computer simulation model requires only a minimum of mathematical knowledge,often avoiding the need for highly trained staff specialists and aiding management understanding of model capabilities and limitations.

A major disadvantage of simulation as a corporate planning tool is the "case study" process by which planning alternatives must be evaluated. Most corporate planning analyses are directed at optimization - i.e., at identifying the most desirable investment and financing alternatives. Using simulation, each computer model solution corresponds to a determination of the implications of a single proposed alternative or specified combination of alternatives. The search for improved plans proceeds via repetitive model solutions, ordered in response to previous results or other insights into potentially desirable alternatives. Sensitivity analysis of model solutions requires similar procedures. Where a large number of corporate planning alternatives and environmental conditions must be considered, the simulation approach implies evaluation of an excessive number of cases, one at a time. In practice, this usually forces a substantial reduction in the number of available alternatives which are actually considered for detailed analysis. Even under such conditions, simulation can be expensive if adequate detail and scope are provided.

Corporate Optimization Models

In contrast to the widespread use of corporate simulation models, few practical applications of corporate optimization models have been reported [9,13]. There are at least several apparent reasons for this:

(1) Optimization implies the existence of a defined planning goal (or goals) which can be formalized in a model. In practice, a variety of different system performance indicators are typically of interest, but these are seldom defined explicity or in a form appropriate for optimization modeling.

(2) Simplifying assumptions about the detailed nature of model variables and relationships are required for many optimizing algorithms, often limiting the attainable level of model realism consistent with computational feasibility.

(3) Optimization modeling may involve different data requirements and planning procedures than those which exist in many corporations.

In addition, the relatively high degree of mathematical sophistication and related technical problems associated with optimization have probably been limiting factors.

Experiences with several operating corporate optimization models indicate that where these problems can be dealt with effectively, the analytical power of optimizing techniques (especially mathematical programming) and available computing software offer significant benefits for corporate planning. Each optimization model solution corresponds to the evaluation of an entire set of planning alternatives and selection of those which best satisfy the defined performance criteria. Efficient sensitivity and parametric analyses of model assumptions and changing conditions are also possible using the same computational techniques. With this approach, a vast number of complex planning alternatives can be evaluated and priorities can be assigned with a fraction of the effort required for equivalent analyses using simulation.

Hybrid Models

Several observations about the simulation and optimization modeling approaches are in order at this point:

(1) Neither modeling approach is ideally suited for use as a corporate planning tool - each suffers from important deficiencies.

(2) Considerable functional complementarity exists between the two approaches. Simulation offers descriptive power and broad applicability but often requires extensive analysis; optimization, on the other hand, offers analytical power but is weaker in descriptive accuracy and applicability. Thus, the strength of one modeling approach complements the weakness of the other.

(3) This suggests that simulation and optimization should be considered as complementary, rather than alternative, corporate modeling approaches. Where possible, it is desirable to exploit the strengths of both through some form of hybrid model or model system.

The incorporation of heuristics or other optimum-seeking routines in computer simulations is one possible avenue toward combined use of simulation and optimization methods in corporate modeling. This suggests use of computerized routines to guide the search for improved solutions, thus reducing the amount of human intervention required in favor of defined, systematic search procedures. Of course, use of an optimum-seeking routine does not guarantee that an optimum will be found, nor does it completely eliminate human judgement from the search process, but it can speed up the search significantly. Optimum-seeking search techniques and their applicability to simulation studies in a variety of contexts have been discussed by others [8,17]. Techniques with particular relevance for corporate financial simulation have also been identified [6].

A more promising approach is to link corporate simulation and optimization models in a corporate model system [6,14,21]. The primary role of optimization in such a system is to search, identify, and screen planning alternatives at an aggregate level. This preliminary evaluation may cover a wide range of possibilities over a multiperiod planning horizon using efficient search algorithms and available computer codes. The outcome is a set of preferred investment, operating and financing alternatives consistent with stated planning objectives and conditions. A simulation model can then be used to project the detailed implications of selected optimization results under specified environmental conditions. Its role in the planning system is to provide a more realistic basis for evaluation of a limited number of promising alternatives, and to test the validity of the optimization model results when more detailed considerations are incorporated into the analysis. Revised planning assumptions and parameters resulting from the simulation can be fed into subsequent optimization runs to refine the outputs. Such a recursive solution approach involving optimization and simulation models has also been found useful in other contexts [19].

This combination of simulation and optimization offers corporate planning support beyond the capabilities of either technique when used alone. In general, optimization provides an overall evaluation of available planning alternatives; simulation is employed to examine those selected in greater detail. Partitioning the overall analysis into "macro" and "micro" stages - with iterations between the two stages - therefore permits use of each technique to its best advantage in the planning process. An operational computer-based corporate planning system which incorporates simulation, optimization, and other analytical models is described below to illustrate this approach.

A CORPORATE MODEL SYSTEM

A major diversified corporation has developed and implemented a system of planning models to improve the efficiency with which alternative combinations of corporate strategies, financing methods, and planning assumptions are evaluated. The modeling system approach was selected to give maximum flexibility in developing a planning support capability consistent with the scope and complexity of corporate-level planning problems. Experience with the system during the past two years has demonstrated both the potential and the operational feasibility of this approach.

Details of the system and its application in a variety of strategic planning studies have been presented elsewhere [13,14]. This discussion will only highlight important system features, including the respective roles of simulation and optimization techniques.

System Components

Following earlier discussions, the key to effective analytical support of the corporate planning process is not a corporate model, but an integrated planning system. The system under consideration reflects the corporate-level focus, financial orientation, and distant planning horizon that characterize strategic planning in most corporations. It consists of five functionally distinct subsystems:

(1) The Information Management Subsystem controls the flow of information, maintenance of the planning data base, and interfaces with data sources, other system components, and users. Explicit provision is made for interactive use of the system, including on-line input preparation, run initiation, and output generation with a wide range of user options. Included in the information management subsystem are system executive routines, the system data base, and conversational input editors and output generators. The input editors organize raw planning data from multiple sources into appropriate data base files, check for arithmetic and format errors, and compare subsidiary projections with historical data and econometric projections to identify questionable estimates. The output generators provide a variety of report options to meet different needs including pro forma financial projections, corporate and planning unit performance summaries, detailed and summary optimization model results, and input error reports.

(2) The Optimization Subsystem selects an optimal set of funds sources and investments subject to a complex set of financial, legal, and operating limitations at both the corporate and planning unit levels. It also permits testing of the robustness of proposed plans and determination of optimal reallocations of corporate resources in response to changes in the planning environment. A large mixed integer mathematical programming model is the basic element of this subsystem, with operating support provided by matrix generation and postoptimal analysis routines. The most operationally effective optimization objective used thus far for planning purposes is maximization of a linear approximation of earings per share over the planning horizon, but other performance measures can also be evaluated with the model. The major planning variables are investment and financing alternatives associated with proposed corporate and planning unit plans. Corporate activities are subject to numerous financial and operating restrictions, some imposed by management policy and others by external forces. Among those represented explicitly in the optimization model are restrictions on the pattern of earnings per share growth, return on assets and equity, corporate funds flow, common financial ratios, short-term debt and stock transactions. A current version of the model contains 700 constraints and 1000 variables, including 250 zero-one variables. Solution of the model using the Univac 1108 computer typically requires 10 cpu minutes to reach the continuous optimum and another 20 cpu minutes to find the mixed integer optimum. Subsequent mixed integer solutions using the previous optimal basis require about 5 additional cpu minutes.

(3) The Simulation Subsystem performs a deter-

ministic financial simulation for predesignated corporate and planning unit strategies. Like most corporate financial simulation models, it operates on accepted financial accounting variables and relationships. Options are provided for evaluating any desired combinations of organizational groupings, corporate and planning unit investment proposals, acquisitions, divestments, and financial strategies. The simulation converts all funds to a common currency, reflects all internal and external funds flows, finances funds deficits from a corporate pool, incorporates proposed acquisitions and divestments, and computes consolidated corporate and planning unit financial statements. The financial consolidation usually requires less than 30 cpu seconds using a Univac 1108 computer.

(4) The Econometric Subsystem provides projections of national and industry economic conditions for use in testing the reasonableness of planning unit projections and in preparing simulation and optimization model inputs. Current econometric support for the system is purchased from a commercially available forecasting service.

(5) The Risk Analysis Subsystem is designed to provide insights into the variability inherent in planning estimates. It currently consists of two models. The Profitability Profile Model is designed to project performance distributions for planning units based on econometric forecasts, historical data, and subjective management evaluations of possible future conditions. These distributions are then used to estimate confidence limits for various corporate profit levels. A pessimistic estimate, called the minimum income level, is derived for every strategy and is incorporated in the optimization analysis. The Business Mix Model applies a portfolio analysis approach to evaluate the risk-return characteristics of the corporate plan and determine corporate asset allocations which maximize expected returns at different risk levels.

System Application

In essence, the subsystems described above constitute a specialized computer-based management information system with extensive analytical capabilities. Its power as a planning tool derives from the integration of diverse, but complementary, planning models with user-oriented information storage and handling features. This integration is accomplished through logical input-output linkages in system operation. Each of the subsystems and component models can be operated independently from the rest of the system, but this is not typically done.

The planning data base is the primary interface between subsystem models and is therefore the most important system link. With few exceptions, subsystem inputs and outputs flow through the data base, where they can be accessed in response to direct inquiry or for further processing within the system. Econometric forecasts, for example, are stored in the data base for input editing and strategy formulation purposes.

Inputs to the Optimization Subsystem include detailed information on proposed planning strategies, available funds sources, and constraint limits. The latter are established by management or are determined through studies involving the Risk Analysis Subsystem models. The investment and financing strategies selected in the optimization analysis are stored in the data base for report preparation and further analysis using the Simulation and Risk Analysis Subsystems. In view of the approximations inherent in the corporate optimization model, of course, these results must be considered as only approximate. Nevertheless, they represent desirable, if not optimal, selections from among the vast number available for consideration. These selected strategies are obvious candidates for more careful evaluation using the Simulation Subsystem. In contrast to the optimization model, the simulation model requires few simplifying assumptions and thus provides a more realistic test of strategy implications. For example, the simulation model computes corporate earnings per share quite precisely for any selected set of strategies; the optimization model can only approximate this figure because of the non-linear effects of expansion and contraction in the common stock pool.

Applications of the corporate modeling system have included a wide variety of periodic and ad hoc planning studies. Periodic studies are conducted at regular intervals (e.g., the annual planning cycle) and typically relate to planning decisions involving the full scope of corporate activity. Ad hoc studies, on the other hand, are conducted in response to problems or opportunities (e.g., an unexpected change in international exchange rates) which require evaluation prior to the next periodic planning review. Periodic studies typically involve all system components and begin with careful input editing and preparation of the system data base. Because of the vast number of alternatives to be considered, the optimization model plays a major role in screening internal investment strategies, proposed acquisitions and divestments, and financing opportunities. Ad hoc studies typically require only minor modifications of the system data base and rely more heavily upon simulation of the implications of particular problems or opportunities.

CONCLUDING COMMENTS

Corporate simulation models will no doubt play an increasingly important role in corporate planning during the 1970's. Applications to date have demonstrated that models can assist in improving both the process and practice of planning in a wide variety of contexts, and the number of models in use or under development is increasing rapidly. However, the vast majority of corporate models in use today are limited to deterministic "case study" simulations of selected planning alternatives. This approach offers advantages in model development and initial implementation, but it can involve extensive computation and effort in the search for improved corporate plans where a large number of alternatives exist. This, in turn, may limit the potential usefulness of simulation models as creative planning tools. Other model characteristics are also important determinants of planning effectiveness and efficiency and must be carefully considered in model design.

The continuing advance of computer and modeling technologies, combined with increasing formalization of corporate planning efforts and growing acceptance of formal planning models, has set the stage for a "second generation" of corporate planning models. One promising direction for evolution is suggested by the combination of simulation and optimization capabilities in a corporate modeling system. A system of models can provide a degree of flexibility and analytical sophistication consistent with corporate planning problems while still preserving the advantages of simulation as a planning tool. Experience with a prototype corporate model system has demonstrated both the feasibility and potential of this approach.

Acknowledgement

The author wishes to acknowledge the significant contributions of his colleague, Michael Moses, who collaborated on the research reported in this paper.

REFERENCES

(1) Ackoff, R.L., A Concept of Corporate Planning, New York, Wiley, 1970.

(2) Agin, N.I., "Corporate Planning Models: What Level of Abstraction?," Proceedings of the 1971 Winter Simulation Conference, New York, December 1971.

(3) Boulden, J.B. and Buffa, E.S., "Corporate Models: On-Line, Real-Time Systems," Harvard Business Review, Vol. 48, No. 4 (July-August, 1970).

(4) Brown, D.B., "Stages in the Life Cycle of a Corporate Planning Model," in Albert N. Schrieber (Editor), Corporate Simulation Models, Seattle: University of Washington, 1970.

(5) Carter, E.E., "An Interactive Simulation Approach to Major Investment and Acquisition Decisions," Boston, Graduate School of Business Administration, Harvard University, February 1970.

(6) Carter, E.E. and Cohen, K.J., "Portfolio Aspects of Strategic Planning," to appear in Model and Computer-Based Corporate Planning, Cologne, Germany: BIFOA International Symposium, March, 1972.

(7) Dickson, G.W., Mauriel, J.J., and Anderson, J.C., "Computer Assisted Planning Models: A Functional Analysis," in Albert N.Schrieber (Editor), Corporate Simulation Models, Seattle: University of Washington, 1970.

(8) Emshoff, J.R. and Sisson, R.L., Design and Use of Computer Simulation Models, New York, MacMillan, 1970.

(9) Gershefski, G.W., "Corporate Models - The State of the Art," Management Science, Vol. 16, No. 6 (February 1970).

(10) Gershefski, G.W., "Building a Corporate Financial Model," Harvard Business Review, Vol. 47, No. 4 (July-August 1969).

(11) Goldie, J.H., "Simulation and Irritation," Supplement to Albert N. Schrieber (Editor), Corporate Simulation Models, Seattle: University of Washington, 1970.

(12) Hall, W.K., "Strategic Planning Models - Are Top Managers Really Finding Them Useful?," Graduate School of Business Administration, University of Michigan, January, 1972.

(13) Hamilton, W.F. and Moses, M., "An Optimization Model for Corporate Financial Planning," Wharton School, University of Pennsylvania, November, 1971; to appear in a forthcoming issue of Operations Research.

(14) Hamilton, W.F. and Moses, M., "A Computer-Based Corporate Planning System", to appear in Model and Computer-Based Corporate Planning, Cologne, Germany: BIFOA International Symposium, March, 1972.

(15) Hertz, D.B., New Power for Management, McGraw-Hill, New York, 1969.

(16) McKenny, J.L., "The Role of Computer Simulations in Planning," in Albert N. Schrieber (Editor), Corporate Simulation Models, Seattle: University of Washington, 1970.

(17) Meier, R.C., Newell, W.T., and Pazer, H.L., Simultation and Business and Economics, New Jersey, Englewood Cliffs, Prentice-Hall, 1969.

(18) Miller, E.C., Advanced Techniques for Strategic Planning: AMA Research Study No. 104, New York: American Management Association, 1971.

(19) Nolan, R.L. and Sovereign, M.G., "A Recursive Optimization and Simulation Approach to Analysis with an Application to Transportation Systems," Management Science, Vol. 18, No. 12 (August 1972).

(20) Schrieber, Albert N. (Editor), Corporate Simulation Models, Seattle: University of Washington, 1970.

(21) Wagle, B., "The Use of Models for Environmental Forecasting and Corporate Planning," Operational Research Quarterly (September, 1969).

A GENERAL UNIVERSITY SIMULATION MODEL

H. K. Eldin Ph.D.
Oklahoma State University
Stillwater, Oklahoma

C. C. Daniel
NASA/MSFC
Huntsville, Alabama

Abstract

This paper presents the results of an attempt to develop a general simulation program for the study of the utilization and allocation of resources in the university environment. The problem area examined covers the use of both physical and human resources. The simulation program was written in GPSS 1100.

Introduction

The use of simulation as a technique for the examination of the interactive aspects of the operation of physical systems has increased greatly within the past few years. The vast majority of these simulations are designed to attack a specific problem and as such are greatly limited in their application.

The lack of generality in simulation programs has tended to foster a large degree of duplication of effort. While many unique situations may exist in which specialized simulation programs are appropriate, there also exist a large number of similiar systems which can benefit from the development of generalized simulation models. The university system is an environment in which the basic elements are relatively stable from institution to institution. The simulation model developed for a general university situation would, with certain modifications, be expected to hold for other universities. Based on this assumption, the authors have attempted to develop a general model for the study of the university resource utilization problem. The results of this effort are the context of this paper.

Description of the Physical System

The program presented in this paper simulates the physical university environment presented in Figure 1. This environment consist of the physical facilities, including the classrooms and the support facilities, and the human resources of the university staff. The basic problem is to examine the allocation and utilization of the available resources under various system conditions.

Each potential class in the basic university system must function within the available student and resource limits. Therefore, of all potential courses to be offered by the university, only those in which there is sufficient student enrollment and for which there are available both the required physical and human resources, can be offered. The enrollment limit for each class level is established by the university on the basis of funds available and alternate demands on the resources. The work time for the university system is also a function of the type of students enrolled in the system. Additionally, this relationship influences both the time and amount of resource utilization.

Desirable Characteristics of the Program

The basis of any generalized simulation program is the ability of multiple users to utilize the program with only minor system changes. To accomplish the objective of program generalization, the following characteristics were included in the program:

1. A relatively simple and easy method of data input.
2. A choice of output options.
3. The ability to vary the discrete control distributions.
4. The ability to change system control limits within a simulation run.

These characteristics tend to provide a program with the flexibility required of a general model.

Model Description

In describing the simulation model, the following areas will be discussed: model assumptions, programming language and computer requirements, unique features , model inputs, model outputs, and model execution.

Model Assumptions

The following assumptions have been made in the development of the university simulation model:

1. Class length is one hour.
2. Any class which does not have a required number of students enrolled or which cannot find available space, will not be offered.
3. Classroom space and instructor personnel are allocated on a first come first serve basis. However, allocation on a priority basis is possible.
4. All control distributions are discrete in nature.
5. The utilization of support facilities, that is, administrative, laboratory, and computational facilities, is separate from the utilization of classrooms and instructors.
6. The daily operating time of the university is variable.
7. Course classrom allocation is made on the basis of the number of students in the course and the classroom range into which the course falls.

Programming Language and Computer Requirements

A general flow diagram of the simulation program is presented in Figure 2. The program is written in GPSS 1100 for the UNIVAC 1108 computer. The number of FUNCTIONS, and MATRIX SAVEVALUES required is dependent on the size of the system to be simulated.

The program is currently set up to run in the 48K partition on the 1108 computer. However, as the size of the system to be simulated is increased, the core required for the program will also increase.

Unique Features of the Program

In developing this simulation model, it became apparent that the greatest problem to be overcome was the requirement for the vast number of transactions generated in the university environment. The approach taken to overcome this problem was the use of a carrier transaction which served the same function as the normal flow transactions. As a result , the initial portion of the program took the following form:

```
*
*  DETERMINE THE NUMBER OF STUDENTS
*  ENROLLED IN EACH SPECIALTY
ZZZ    ADVANCE
SPECNU VARIABLE X$TOTNU*FN$SPEC.PROB
       MSAVEX SPEC1(P$PAR1,1),V$SP-
       ECNU
       LOOP PAR1,ZZZ
*
```

A single transaction performs the calculations for all specialities in the system. The enrollment in each speciality is determined on the basis of a discrete probability function inputed at program initiation.

Another unique feature of the program is that all program inputs are provided by either the INITIAL or the FUNCTION statements. A description of the input procedure will be provided in a later section of this paper.

To provide for multiple simulations of a number of semesters,the program can be utilized with a RESET card as shown below:

```
*
       START 1
       RESET
       START 1
*
```

A system printout will be provided for each START card. Any number of basic simulations may be run in this manner.

Model Inputs

As stated previously all model inputs can be provided through the use of the INITIAL and FUNCTION cards.

It is assumed that sufficient data is available to provide for discrete distributions in the following areas:

1. Total number of students enrolled.
2. Percentage of students at each class level, i.e., junior,senior, etc.
3. Percentage of students in each speciality.
4. Percentage of each type of student, i. e., full time, part time, etc.
5. Distribution of class types and sizes.
6. Distribution of course size requirements.
7. Distribution of instructor requirements.
8. Distribution of support requirements.

The INITIAL card inputs involve such factors as:

1. Class size limit.
2. Work hour limits.
3. Classroom definitions.
4. Instructor definitions.
5. Speciality course definitions.

The program output requires a MSAVEX, matrix savevalue, definition for each classroom. The definition takes the following form:

```
     *
     *  INSTRUCTOR DEFINITIONS
            MATRIX SMITH(8,5),JONES(8,5)
     *  CLASSROOM DEFINITIONS
            MATRIX EN315(8,5),EN215(8,5)
     *
```

At the beginning of each simulation, the instructor and classroom schedule are reset to zero.

```
     *
            INITIAL SMITH(1-8,1-5),0/
                    JONES(1-8,1-5),0
     *
            INITIAL EN315(1-8,1-5),0/
                    EN215(1-8,1-5),0
     *
```

The possible courses which may be offered are also supplied by INITIAL cards.

```
     *
            INITIAL INDEN(1,1),4133/
                    INDEN(2,1),4211
     *
```

Classroom size limits are specified in range values as follows:

Range	Size		Cummulative Freq.
1		10	.05
2	10	20	.45
3	20	50	.65
4	50	100	.85
5	100	250	1.00

The discrete function is then inputed to the program in a FUNCTION statement as follows:

```
     *
     RANGE    FUNCTION,D V$CALC,.05,1
              .45,2 .65,3 .85,4 1.0,5
     *
```

The major portion of the simulation model consist of a schedule update sequence based on a series of indirect assignment blocks. The assignment of a classroom name takes the following form:

```
     *
                  ASSIGN PAR17,*CLASS.NAME
     *
```

where CLASS.NAME is a name valued function containing the utilization distribution for the classrooms.

The hour at which the classroom is to be used is specified as follows:

```
     *
                  ASSIGN PAR16,FN$TIME
     *
```

where TIME is a numeric valued function containing a discrete distribution relating to the hours of the day during which the classroom will be available.

Model Outputs

The outputs from the simulation program are composed of the following:

1. A master schedule for each speciality showing:
    a) Course number.

b) The instructor who is teaching the course.

c) The credit hours for the course.

d) The number of students enrolled in the course.

e) The classroom utilized.

f) The time of day at which the course is to be offered.

2. A schedule for each classroom.
3. A schedule for each instructor.
4. A listing of all nonscheduled courses.
5. A listing of all request for support services.

This output may be restricted or expanded through the use of the PRINT option.

Model Execution

The program has been setup to simulate on a semester basis with the number of semesters to be simulated as a user option. The simulation takes the form of multiple schedule modifications as follows:

```
*
*  UPDATE CLASSROOM SCHEDULES
       MSAVEX  *PAR17(P$PAR16,1),P$-
               PAR7
       MSAVEX  *PAR6(P$PAR5,6),P$PA-
               R16
       MSAVEX  *PAR17(P$PAR16,5),P$-
               PAR7
*
```

The use of indirect referencing allows for the use of any number of classrooms without model modification. The instructor schedules are updated in a similiar manner, however, the instructor selection process is based on a specialty selection process. An example of the specialty selection process is as follows:

Type	Instructors
1	Jones, Smith, Eldin
2	Jones, Brown, Daniel
3	Brown, Smith
4	Eldin, Brown

The selection sequence for the above inputs is as follows:

```
  *
             ASSIGN PAR18,MX$*PAR6(P$PA-
                    R5,2)
             ASSIGN PAR19,*INST.FUNCT
             ASSIGN PAR30,FN$*PAR19
             ASSIGN PAR20,*INSTRUCT
  *
```

The above sequence will select an instructor of the required type and will ready his schedule for update. All outputs are in the form of name valued attributes. This option provides for easy simulation analysis.

Analysis

The university simulation model was developed in order to study the interactive aspects of the university scheduling and resource allocation problems. The model was designed in such a manner that system input data and model operating sequence could be altered without an alteration of the basic model itself. In the process of developing the model the following options were considered:

1. A calculation of the utilization of the university personnel and facilities.
   This option was factored into the model, however, the feasibility of calculating a utilization rate for university personnel was found to be limited. This limitation developed due to the multiple aspects of the instructor's work load.
2. A projection analysis and forecasting system was examined for use in the area of instructor and classroom definitions.
3. Several methods were examined for use in the study of the support and service facilities of the university. The method which was finally factored into the model provides the following data:
   a) A listing of all request for support along with the type of support requested.
   b) A tabulation of the number of request for each support facility along with the durations of the support activity.

The procedure for comparing two alternative systems is based on simulation runs utilizing the same random number base. The system outputs may be compared on either a direct utilization basis or on the basis of the system scheduling sequence.

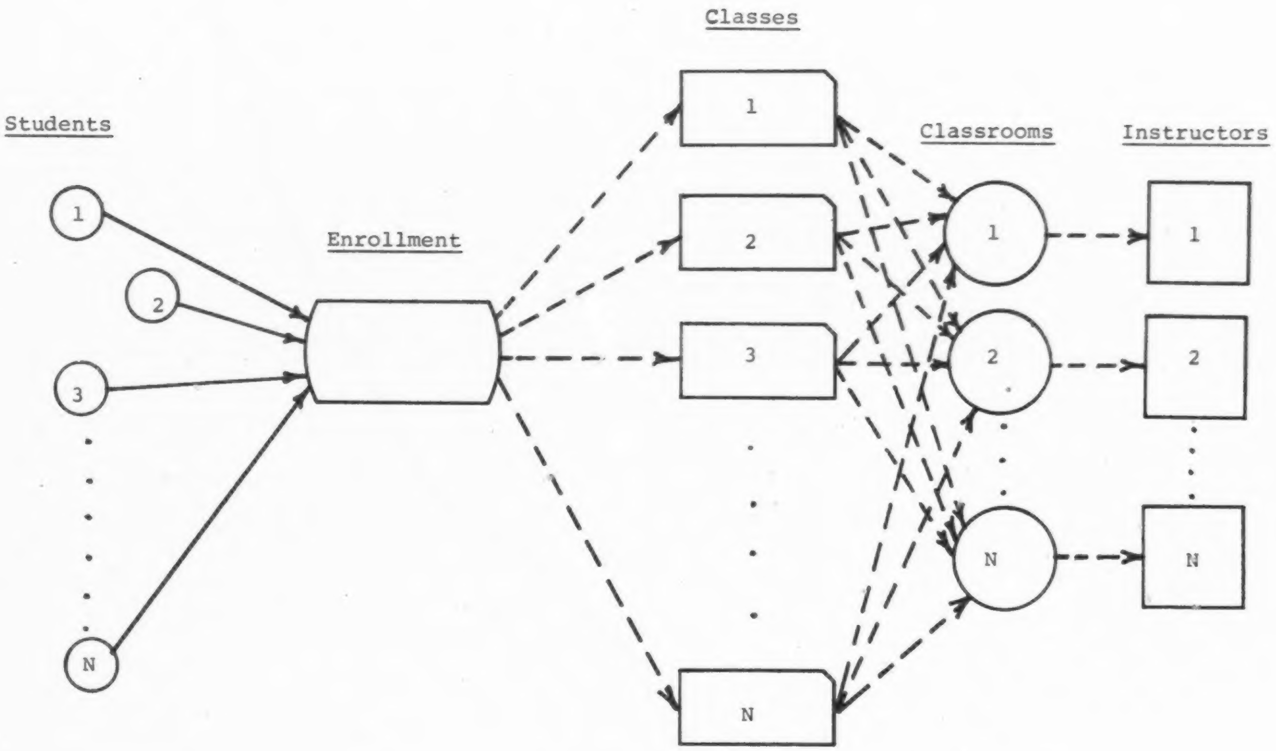
Conclusions

The university simulation model presented in this paper provides the basic structure for an effective method of analysis of the dynamic university system. The system outputs are provided in a form which can easily be analyzed.

The main limitation of the program is in the area of the required supporting discrete distributions. If sufficient data is available the basic model should be applicable to any university system.

When a large system is to be simulated it is recommended that the user utilize a data tape for inputs rather than standard punched cards.

The model presented in this paper is intended to illustrate that the technique of simulation can provide a valuable tool in the study of the university system. While for the factors considered the basic model does provide a general simulation basis, there are certain unique system characteristics which may be required in each application. It is believed that the options provided in the simulation program will allow for the consideration of these unique features without major program changes.


Students
1
2
3
N
Enrollment
Classes
1
2
3
N
Classrooms
1
2
N
Instructors
1
2
N

The number of classes, classrooms, and instructors can vary from 1 to N. The flow pattern is a user option.

Figure 1. The University System Configuration

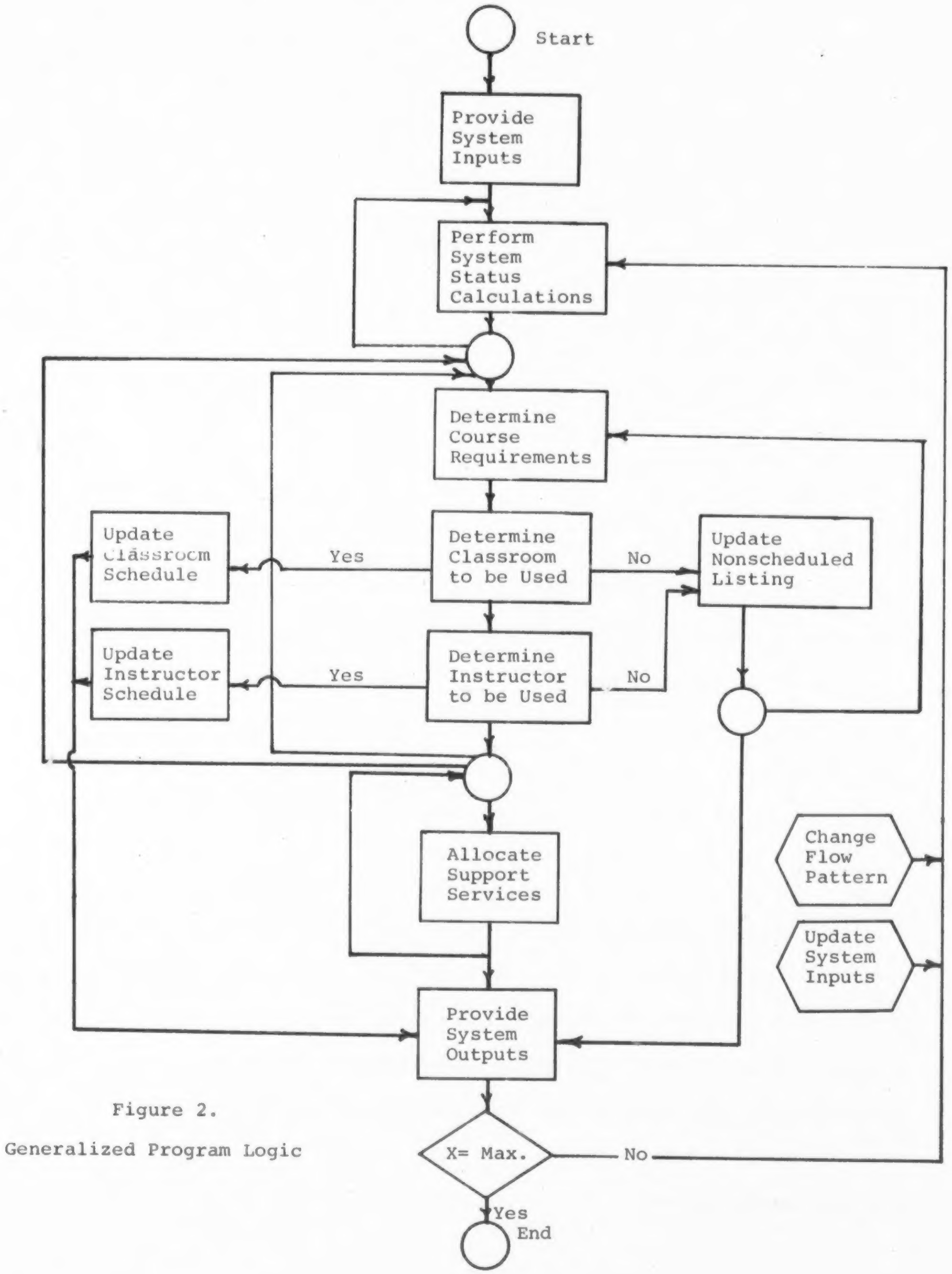

Start
Provide System Inputs
Perform System Status Calculations
Determine Course Requirements
Update Classroom Schedule
Yes
Determine Classroom to be Used
No
Update Nonscheduled Listing
Update Instructor Schedule
Yes
Determine Instructor to be Used
No
Change Flow Pattern
Allocate Support Services
Update System Inputs
Provide System Outputs
X= Max.
No
Yes
End

Figure 2.

Generalized Program Logic

RATE-REVENUE-COST OF SERVICE SIMULATION OF A NATURAL GAS UTILITY

D. Jeffrey Blumenthal

On-Line Decisions, Inc.

ABSTRACT

A major midwestern utility was faced with the dilemma of increased demands for natural gas energy while the supplies of natural gas were diminishing. Questions on how to meet seasonal or off-peak demands, how to utilize underground storage, and how much new supply would be required to eliminate waiting lists for residential customers were typical of the problems confronting this utility. In addition, the cost of service due to inflation and pipeline cost adjustments indicated that this company should perhaps request a general rate increase and required substantial calculations in the PGA (Purchase Gas Adjustment) area.

A set of linked models was built for this utility to answer these questions. The models considered the areas of rate, revenue, cost of gas, PGA, and income. Each model could be run independently of the others; however, all could be linked to provide an integrated planning system for the utility. The models were implemented in a conversational mode on a time-shared computer and allowed the model users to interactively vary rate structures, pipeline contract terms and alternative sources of supply over a five-year planning horizon.

INTRODUCTION

Historically, natural gas has been considered as an inexpensive source of energy. In addition, it is virtually nonpolluting. As a consequence, the demand for natural gas energy has reached all-time highs. The traditional peak demand period for natural gas has been the winter space heating season. However, because of the non-polluting aspects of natural gas energy, electric power generating companies have placed off season or summer demands on natural gas energy. Many companies have constructed under-ground storage facilities for servicing the peak winter heating season demands. Most natural gas utilities, therefore, have a variety of alternatives on how or to whom they should sell natural gas. The supply of natural gas, on the other hand, has remained relatively constant for the last several years, resulting in an apparent shortage of natural gas in the past several winter heating seasons. New supplies or potential new sources of natural gas appear to be considerably more expensive than existing ones. New processes, such as coal gasification, represent a potential energy source. However, their impact is still several years off. A proposed pipeline from the north slope of Alaska would represent a considerable capital investment on the part of a pipeline company. Liquefied natural gas from the Middle East has been suggested for alleviating some of the demand requirements on the Eastern Seaboard. No doubt a combination of all of these alternatives is possible. Investor-owned public utilities are regulated either by the Federal Power Commission or by a statewide organization. Servicing an area rapidly growing in population, a major midwest natural gas utility was faced with finding supplies of new, more expensive gas. The company was also involved in major investments in underground storage facilities. With increased investments in plant and equipment, would a rate increase be justified? How fast would the company recoup additional costs of the more expensive gas? What new supplies would be required to eliminate waiting lists and satisfy the off-peak demands of power generating companies? The answers to these questions were not readily available. Moreover, in seeking answers, the company uncovered an additional problem--

that of varying internal forecasts. Marketing based its figures on one set of assumptions, while Operations worked on yet another. The Treasurer's group at headquarters was assigned to reconcile these forecasts into a unified operations plan. In spite of earlier failure at modeling, the company felt that simulation models could be useful in consolidating these forecasts and developing a unified operating plan which would answer questions concerning alternative supply sources, sales to customers on waiting lists, and off-peak demand sales.

OVERVIEW OF THE MODEL SYSTEM

Most utilities normally function in four operational areas. These are:

1. Rates and revenues
2. Operating costs
3. Construction and capital budgeting
4. Financing

A fifth area, that of consolidation, is also necessary. Ogden (Reference 1) discussed the problem of modeling these particular areas and illustrated the types of questions and data required.

The needs for the company under discussion here, however, and for most operating gas utilities today, is in the areas of rates and revenues and the operating costs. These areas require additional detail to answer "what if" questions, necessary for computer simulation. Consequently, a top level planning system was developed for this utility illustrated in Figure 1, and consisting of five separate operating modules.

The basic design was developed during a one-month initial Phase I study, consisting of interviews with key executives within the organization. Data availability and traditional forecasting methods were also studied at this time. The result of Phase I was the five module system to simulate the rate and cost of service area for this utility by months over a time horizon of five years.

The Rate Module isolated the smallest definable element of revenue to the company. Customers were classified by the rate schedule utilized, their particular revenue class, such as residential space heating, and by billing method. The primary purpose of the Rate Module was to determine the actual therms sold and the base dollars of revenue generated in each of these elements.

The Revenue Module took the therms sold from the Rate Module and determined the therm sendout--the actual number of therms distributed in a particular month. The sendout is related, but not equivalent to the therms actually billed. In addition, the revenue model added various adjustments such as purchased gas adjustment (PGA) and various state and municipal taxes to the base dollar revenue.

The Cost of Gas Module was primarily concerned with pipeline contract purchases and various injections and withdrawals from the company's own underground storage. The company utilized several methods of inventorying storage gas. The model valued inventory to arrive at a total gas distributed figure. The model, of course, evaluated precise pipeline purchases and the cost of those purchases to be used by the PGA model. New, as well as existing sources, were considered.

The PGA module duplicated the actual PGA calculations. The cost of purchases for twelve of the past thirteen months was compared to a base cost. The difference was amortized by formula to a rate per therm. Six major PGA rates were calculated in this module. These rates were then fed back to the Revenue Module for calculation of total dollar revenue.

The Income Module served to consolidate the revenue and cost dollars to arrive at a simple income statement. Some of the outputs were an earnings statement and a return on the rate base.

TECHNIQUES USED IN MODEL CONSTRUCTION

As mentioned earlier, one of the basic outputs of the Rate Model was the forecast of therm consumption. For certain elements, that is, rate-revenue class-bill basis, only a small number of customers were involved and here forecasting therms sold was done by simple extrapolation of past data. In a preponderance of cases, however, a large number of customers would be represented by a particular rate element. Here, therm use was forecast by breaking out the average use per customer and multiplying by the number of customers.

This breakout of the number of customers was extremely useful in simulating a waiting list of various types of service. The model user could vary the number of customers to suit the availability of gas.

Forecasting the average use per customer, on the other hand, was more difficult. In the majority of cases, the amount of usage by a particular customer is variable both with the respect to time. In fact, the largest class of natural gas users were those whose usage depend heavily on weather, i.e., space heating customers.

Figure 2 illustrates the relationship between daily therm usage and outside temperature for a typical residential space heating customer. The figure illustrates the relatively nonlinear behavior of usage versus temperature. A simple transformation of data allows the curve to be broken into two approximately linear sections. This is done by defining what is known as the degree day. By performing a linear regression of usage against degree days, one can determine the base use as the intercept, and the slope as the use per degree day. Figure 2 illustrates this regression line.

In actual fact, however, the problem is not quite that simple. As can be seen in Figure 2, breaking the curve into two linear segments does not match up over the entire range. The basic departure of the regression line from actual usage occurs at both ends of the temperature curve.

To isolate temporal effects from the basic weather effects, the linear regression of average usage per customer as a function of degree days was performed on a twelve month rolling basis. A regression was performed for twelve months points, then the oldest term was dropped out and a new term added. By repeatedly doing the regression, one could take the resulting figures for base use and use per degree day and extrapolate trends in their growth. The model allowed the user to assume various growth rates in both the base use and the use per degree day.

To calculate the base dollars of revenue was perhaps the most complicated of the forecasting formulas used in the model system. A typical rate schedule as illustrated by Figure 3, explains the techniques used for the majority of cases which depicts a space heating rate.

Shown in figure 3 is the base dollars of revenue to a particular customer, based on the total amount of therms that customer uses within a particular month. The figure illustrates that the rate schedule consists of a series of piece-wise linear segments, each one known as a block. Since a straight line can be represented by two points, an intercept and a slope, it is possible to relate the dollars of base revenue as a function of therm within any particular block by such two numbers. If all customers utilized the same amount of gas, the base dollars of revenue could be simply calculated by taking the revenue per customer times the number of customers, where the revenue per customer could be computed from a fixed plus a variable component, depending on usage multiplied by the number of therms. This, of course, can be easily recognized as a simple linear regression. Indeed, regression is often utilized for calculating basic revenue as a function of average therms used. However, all customers don't utilize the same amount of gas. Figure 4 illustrates bill density, basically a distribution of the number of customers utilizing a specific number of therms. The particular shape of the curve, of course, varies from month to month and depends a great deal on the type of service offered. Bill density, or bill frequency analysis, as it is sometimes called, is performed by most utilities. Consequently, curves of the type in Figure 4, are often readily available. Once such a curve has been defined, it is possible to calculate the base dollar revenue. If

(3.2) $f(t)dt$ = number of customers using t to $t + dt$ therms then, if we let $r(t)$ represent the revenue from the rate structure, (3.3) $r(t)$ = base revenue/customer using t therms

then the revenue can be calculated as

(3.4) $$\text{Revenue} = \int_0^\infty r(t)\ f(t)\ dt$$

While very attractive from a theoretical point of view, Equation 3.4 is not very useful from a practical point of view. For one thing, an integral of infinite range is quite hard to simulate.

One's first impulse, is to limit the range of the integration. However, virtually any limitation usually excludes the one oddball customer extremely far out on the usage curve, and it is precisely this customer generating high use, who represents large dollar revenues.

To get around this problem, an alternative formulation can be developed. This is done by considering the percentage of the total number of therms which are sold in a particular block. Since all but the last block end at a finite number of therms, it is possible to calculate the total number of therms used in all, except the last block. The remainder, of course, is then allocated to the final block so that the infinite integral never has to be evaluated. To calculate the number of therms used in a particular block, one defines what is known as an ogive. The ogive, or commodity distribution, is defined by

(3.5) $$G(x) = \int_0^x t f(t)dt + x\int_x^\infty f(t)dt$$

$G(x)$ represents the total number of therms sold in the blocks 0 to X therms. It consists, basically, of two parts, the first part containing all those therms sold to customers whose usage terminated in the block 0 to X. This is represented as the integral from 0 to X of a particular therm level times the number of customers in that level integrated from O to X. For customers whose bills terminate beyond the block O to X, their total usage is simply X times the number of customers whose bills fall outside of that range. Although the second integral appears to be of infinite range, it should be recognized that the number of customers whose bills do not terminate between O and X therms is equal to the total number of customers less those whose bills do, in fact, terminate between O and X. Consequently, one can avoid performing the infinite integration. The ogive corresponding to the bill density illustrated by Figure 4, is shown in Figure 5. The use of ogives is discussed in most introductory texts on rate making fundamentals. However, a particularly good treatment, one which is correct in mathematical terms, is given in Reference 2. There are several normalizations which can be performed on the bill density and ogive distribution. The first of these normalizations is to express the number of customers or number of therms; that is, representing the functions $f(t)$ and $G(x)$ as a number ranging between O and 1. This normalization makes the bill density and ogive curves correspond roughly to probability densities and distributions. A second normalization used is to translate the therms used by a particular customer and relate them to the average for the entire therm usage. As an example, a customer utilizing 150 therms in a particular month, where the average customer utilized 75 therms, would be normalized to a therm usage of 150 divided by 75, or 2. A rather surprising result of this normalization is to remove the effects of weather from the ogive and bill density. The utility under study constructed distributions for each rate and revenue class on a monthly basis. The advantage of using an ogive is obviously that one can examine both changes in a particular rate in a particular block or the changing of individual block sizes. Such analyses are essential for any rate case preparation. The model constructed for this utility could, of course, examine various rate changes and prepare the basic data needed for rate analysis. However, ogives prepared on past data are only history. If one can assume that basic distribution of customer usage remains constant, then utilizing an experimentally observed ogive may be fine for the forecasting problem. However, where it is expected that the density will change its shape; that is, skew one way or the other, depending on the types of customers brought on, then it is necessary to use techniques other than the experimentally observed ogive to calculate the revenue. A technique utilized in this model was to take a curve such as illustrated in Figure 4 and represent it by a mathematically defined probability density function. For instance, Figure 4 was found to be similar to the gamma-1 probability density function (Reference 3). The gama density is a large family of two parameter distributions. Since the bill density is normalized to unit mean, a two parameter distribution will have one remaining parameter, the variance, for adjustment. By changing the variance, the gamma distribution could be skewed to the right or left and vary its shape over a considerable range.

The company's existing bill frequency analysis program was modified to plot curves similar to Figures 3 and 4, as well as giving the normalized ogive at 100 selected points and calculating what value of the variance to use in the gamma-1 density function. In addition to the gamma-1 density, a pareto distribution and several single parameter density functions were also made available to the program, including exponential, uniform and the triangular density functions.

With this added flexibility, a model could be used not only to analyze changes in the rates and the blocking size, but also examine the effect of various distribution changes in the types of customers utilizing a particular rate. Despite the relatively large number of assumptions and smoothings of data, beginning with the degree day-average use relationships and culminating in the use of the gamma function to express customer density, the overall error, in both therms forecasted and basic dollars revenue for the basic revenue elements, did not exceed one half of one percent in total.

The therms used by each of the revenue elements served as a basic input to the revenue model, whose first function was to calculate actual therm sendout. For customers billed bimonthly, a bill would represent sendout extending back into the past two months. If one assumes that customers are billed uniformly throughout the month, one arrives at what is commonly known as the 25-50-25 rule, indicating that, of the therms billed in a particular month, 25% of them were sent out in that month, 50% from the previous month, and 25% in the month prior to that. For monthly billing basis customers, the rule is 50-50. Certain customers' usage could be identified exactly in the month in which it was sent out. These are typically major power generation companies whose meter is read at the end of a particular month, every month, and the sendout corresponds exactly with billed therms.

The cost of gas module was used to evaluate pipeline purchases and cost out the dollar value of service for a particular accounting period. It also provided a forecast for pipeline purchases to be used in the purchased gas adjustment calculation. At the present time, all pipeline purchases are done on contract, using what is known as a two-part rate schedule. The cost to the utility of a particular quantity of gas is broken down into demand and commodity charges. The demand charge is based on the maximum allowable daily draw of gas from a particular pipeline, whereas commodity charge is a variable charge, depending on the number of units withdrawn from the pipeline. Stiff penalties are also included in the rate. These penalty charges prevent a particular utility from withdrawing a greater quantity of gas than that specified in the demand charge.

The net effect of the two-part pricing forces the utility to withdraw virtually all of the demand quantity gas. Doing this achieves the lowest cost per unit figure. The ratio of actual pipeline withdrawals over the contract demand quantity is known as the load factor. Most utilities operate at a load factor approaching very nearly 100%. By utilizing underground storage facilities, it is possible for companies to maintain a virtual 100% load factor, selling what gas they can to customers and pumping the remainder in or out of storage, as the case may be.

The cost of gas module handled two factors involved with the underground storage. The first, that of inventory, was accounted on a layer by layer basis, using the LIFO method. The model could duplicate the accounting relationship required to perform the LIFO evaluation. However, company-owned underground storage was lumped into one massive underground pool, whereas, in fact, the company, itself, utilized six separate underground storage facilities. The model would take the sendout coming from the revenue section, compare it with pipeline purchases, and compute a net injection or withdrawal figure for underground storage. This figure was compared with guidelines used to establish maximum injection and withdrawal rates from underground storage. Needless to say, however, since all storage fields were lumped into a single storage field, considerable judgment was allowed on the part of the model operator on whether or not such injection or withdrawals were indeed realistic or even physically possible.

One additional feature of the model was its ability to perform the energy and pressure adjustment factors. Most pipeline withdrawals and injection rates were computed in units of thousands of cubic feet of gas purchased, withdrawn, or injected in storage. Gas, however, is normally sold on a therm or energy content basis.

Consequently, the model would adjust gas pressure to normal athmospheric pressure and adjust the energy content based on the number of therms per thousand cubic feet. Although of not great significance, the energy content of gas has varied a few percent during the past several years. Needless to say, the computer simulation model performed this annoying calculation without very much trouble. The purchased gas adjustment model performed an involved calculation used to adjust the cost of gas above a given established cost. If, for example, natural gas from the pipeline cost $.07 per therm, and $.05 per therm was the established base price, then the utility was allowed to charge a basic $.02 per therm PGA or purchased gas adjustment. The PGA allows the public utility to keep pace with pipeline price increases. Historically, most pipelines and utilities have been heavily regulated so that prices, themselves, change only infrequently. However, pipelines have an adjustment factor similar to those in effect by the public utility. Hence, changes in price from the pipeline can occur almost daily and a month does not go by where there is no change in the PGA. The actual PGA calculation, itself, takes the established purchases from the cost of gas module and assumes a 100% factor to the pipeline. That is to say, the consumer of the gas does not become penalized if the utility does not utilize 100% of its available demand gas.

Month to month variations are smoothed out by considering twelve of the past thirteen months. The cost of this period is averaged and compared to the base rate figure. The difference then goes into the PGA rate per therm to be sent to the revenue model. To handle the effect of pipeline increases occuring in midmonth, a spreading factor was used to establish a modified PGA rate. Six separate PGA rates were computed by this model and passed to the Revenue Model. In addition to calculating sendout, the Revenue Model added the previously calculated PGA rate times the number of therms sold to the base dollars of revenue. Because the Revenue Model considered therms billed, it was necessary to allocate, using a spreading formula, the PGA rate over the past several months to compute an average PGA rate for a particular rate and revenue class.

Taxes, such as municipal service taxes, proportional to revenue dollars, were also computed in the revenue model. The revenue dollar figure was then passed over to the Income Model.

The basic purpose of the income module, simplest in this system, was to consolidate the revenue dollars and the cost dollars, adding other factors to get an income and earnings per share and a return on the rate base. The income model served primarily as a report generator.

Income taxes, per share earnings, and return on the rate base were virtually all the calculations performed in this particular model. Many of the factors were left as inputs. Major areas not included in the income model were the capital budgeting, capital expenditure area and the financing area.

IMPLEMENTATION AND VALIDATION

The system of models was implemented in Fortran using the GPOS package of On-Line Decisions, Inc. and runs on a time-shared computer in a highly conversational manner. There were several reasons for choosing this type of an operating environment: the scientific-algebraic nature of the revenue, bill frequency calculations, as well as the availability of a large number of subroutines and subprograms for data referencing in GPOS. Though Fortran appeared to be the best language, programming was not a key factor. Since they were based on an accounting system, the models were relatively simple in terms of discreet events. In accounting, closing the books occurs once per accounting period. The time horizon of the model was five years by months. Consequently, the models, themselves, will run 60 times, once for each month of the five-year planning horizon. Fortran subprograms on the On-Line Decisions' Operating System took care of the time variation in the data.

Time sharing was chosen for two basic reasons: to insure availability of the operating system, and to heighten the degree of interaction required to run a particular simulation model.

The models had very few decision rules programmed into them since they were not optimizing models. The project's goal was to allow middle management to actively interrogate the models to answer "what if" questions. In this instance, the model builders and model users were different individuals. Because of their highly interactive nature, the models were easy to build, but hard to run. The person who ran the models was required to interpret the results and decide on an appropriate course of action, i.e., modify the input data appropriately. Because of the dichotomy between builders and users, considerable effort was put into designing the appropriate interface for the nontechnical user.

The GPOS package handled most of the conversational programming within the model system. To validate the model, it was decided that two years of actual data would be placed into the system and the results compared with actual results. In the PGA area, it was noted that arithmetic errors had been made in certain instances. Once detected, accounting would input reconciling items to offset these previous calculation errors.

Needless to say, to model the randomness of human error making was a difficult task. Provisions were made, however, to include reconciling items in many of the key areas. All told, the process of validation took approximately twice as long as the total programming and implementation phases.

ACCEPTANCE AND USE

After model validation, the model data collectors had to switch hats. Instead of concerning themselves with data collecting and analysis, they now had to consider forecasting and formulation of alternative strategies to the model.

To help gain an understanding of the key and critical relationships, a sensitivity analysis was run over most of the variables within the model system. This analysis involved placing small changes in the input variables and noting the effect on key model outputs.

The GPOS system had this sensitivity capability already programmed within it. After initial forecasts and alternatives had been run through the models, a shift in emphasis in the models began to be observed. For example, since the PGA module duplicated the hand calculations used in the PGA calculations, the module began to be employed by the people within that section to check their own calculations.

In addition, although the model did answer the questions concerning the rate-revenue-alternative source of supply questions, the need for other areas soon became apparent.

While top management had initiated the project, they were not involved in the model construction and validation phase. With the introduction of the working model, "what if" questions and alternative strategies were initially slow in coming. However, usage of the system has averaged approximately 30 hours per month on a connect hour basis. The users fall into approximately three categories - at the top level, the senior financial officer; at the middle management level, the assistant treasurer; at the staff level, within the financial department, various planner analysts.

CONCLUSION

From the time the concept of simulation modeling had met with initial acceptance to the time when the completed modeling system was accepted for use, approximately five months had elapsed. Four men worked almost continously on the project.

The first month of the project was spent performing a feasibility study, specifically indentifying the key areas for modeling, the people who would be involved in the modeling project and the types of "what if" questions that needed to be answered.

Phase II required four months to complete with approximately half of this time spent in technical specification of the modeling system. The specification period went through existing forecasting methods, analyzed data availability techniques to be used as to their accuracy and validity, organized the way in which data would be input to the model, and the report formats coming from the modeling system. The programming phase of the modeling project required about three weeks to complete and validation, almost six weeks to complete. The whole exercise of validation was viewed as a training course for the planner analysts involved in the collection and usage. At the end of the four month period, a course was run to review with middle management and to present to top management the techniques and results of the previous Phase II work.

One year has lapsed since the model was accepted by this utility. The model answers a variety of "what if" questions almost daily. The personnel within the assistant treasurer's staff are becoming known as "keepers of the model," and this staff is being given more and more responsibility in determining and accepting strategies in meeting future operational planning.

Although the model has been in use for an entire year, the model is not frozen. . .a favorable indication. Indeed, a planning model should be dynamic and adjust to changing planning conditions. The true benefits of this simulation modeling system are just now becoming evident. Coal gasification plants are being constructed and arctic pipeline contracts are being negotiated. With the model, utility officers can examine how these new sources of natural gas energy will effect the company's operation and insure the stockholders an adequate return on their investment.

LIST OF ILLUSTRATIONS

Figure 1 OVERVIEW OF MODEL SYSTEM

Figure 2 THERM USAGE AND WEATHER

Figure 3 TYPICAL RATE STRUCTURE

Figure 4 BILL DENSITY

Figure 5 OGIVE DISTRIBUTION

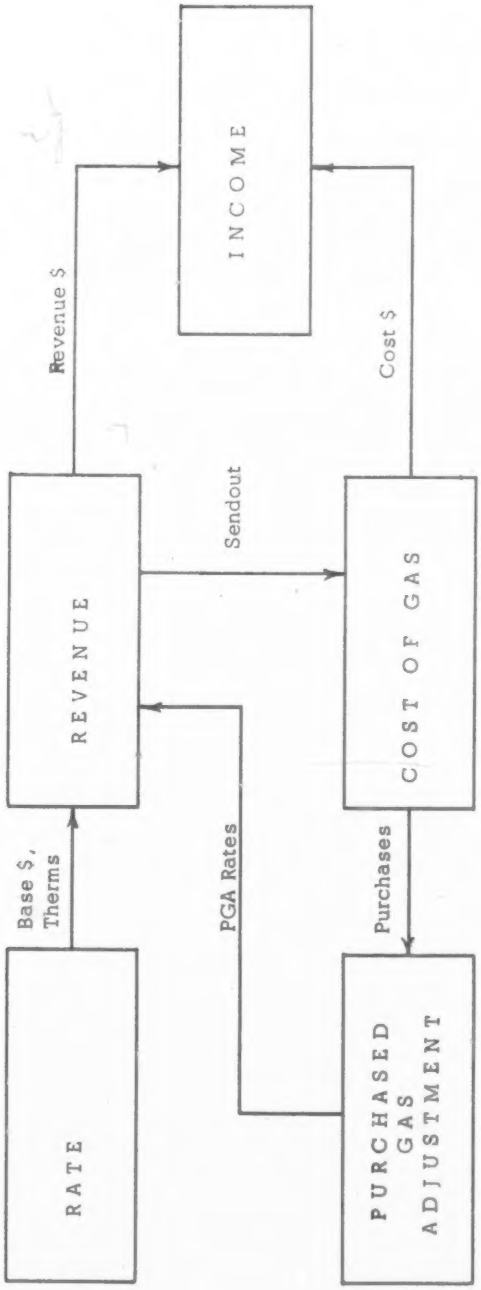

RATE
Base $, Therms
REVENUE
Revenue $
INCOME
Sendout
PGA Rates
COST OF GAS
Cost $
Purchases
PURCHASED GAS ADJUSTMENT

FIGURE 1

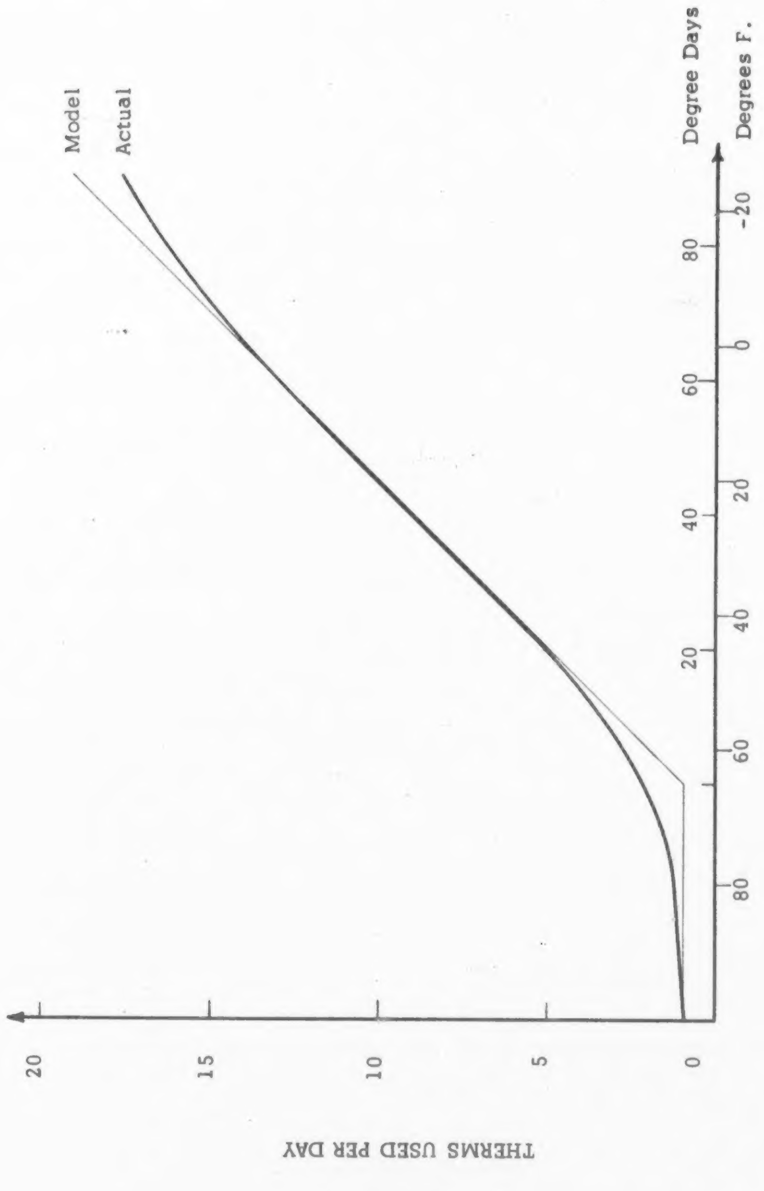

Model
Actual
Degree Days
Degrees F.
20
40
60
80
80
60
40
20
0
-20
0
5
10
15
20
THERMS USED PER DAY

FIGURE 2

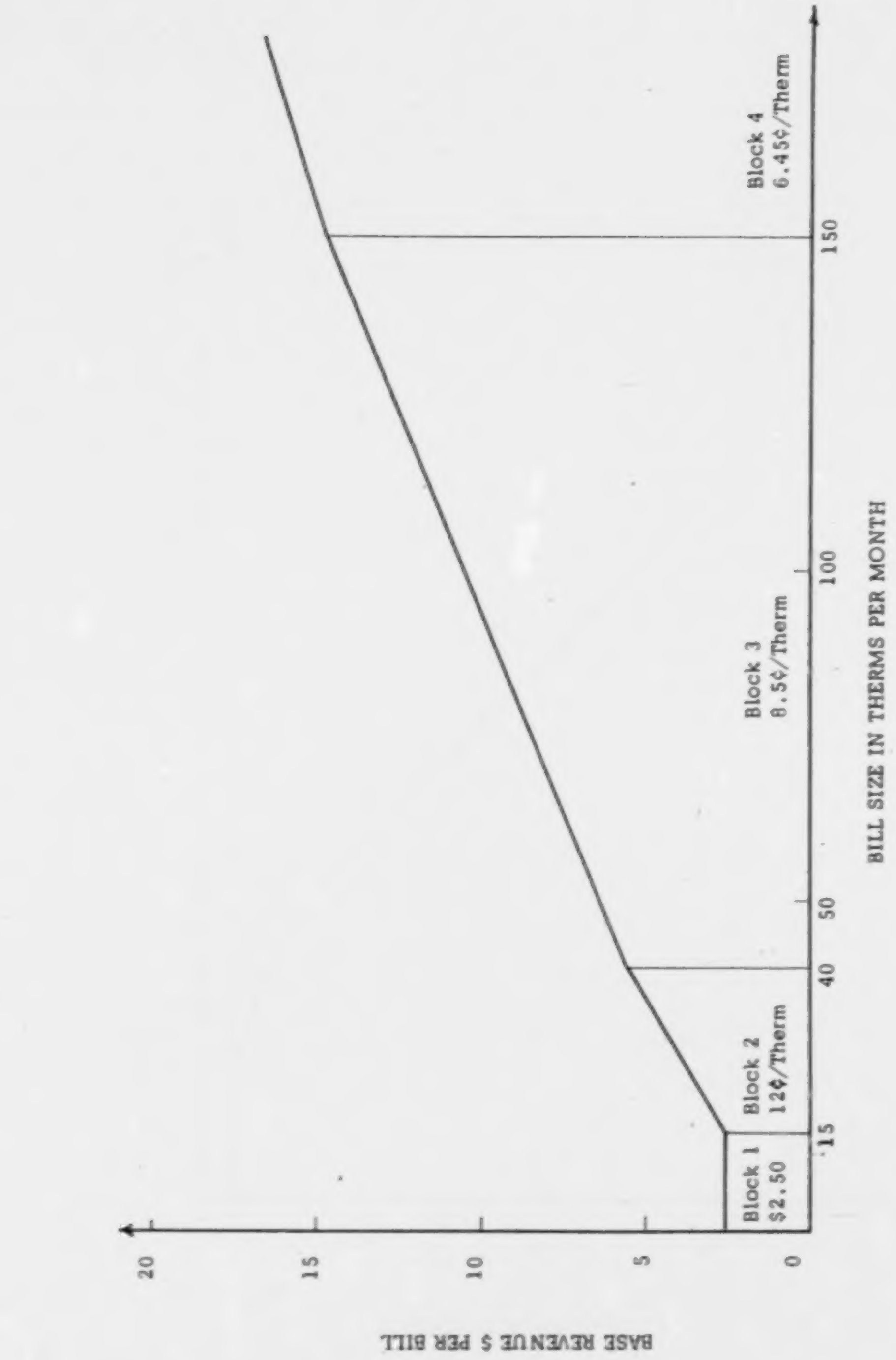

BASE REVENUE $ PER BILL
0
5
10
15
20
BILL SIZE IN THERMS PER MONTH
15
40
50
100
150
Block 1
$2.50
Block 2
12¢/Therm
Block 3
8.5¢/Therm
Block 4
6.45¢/Therm

FIGURE 3

9

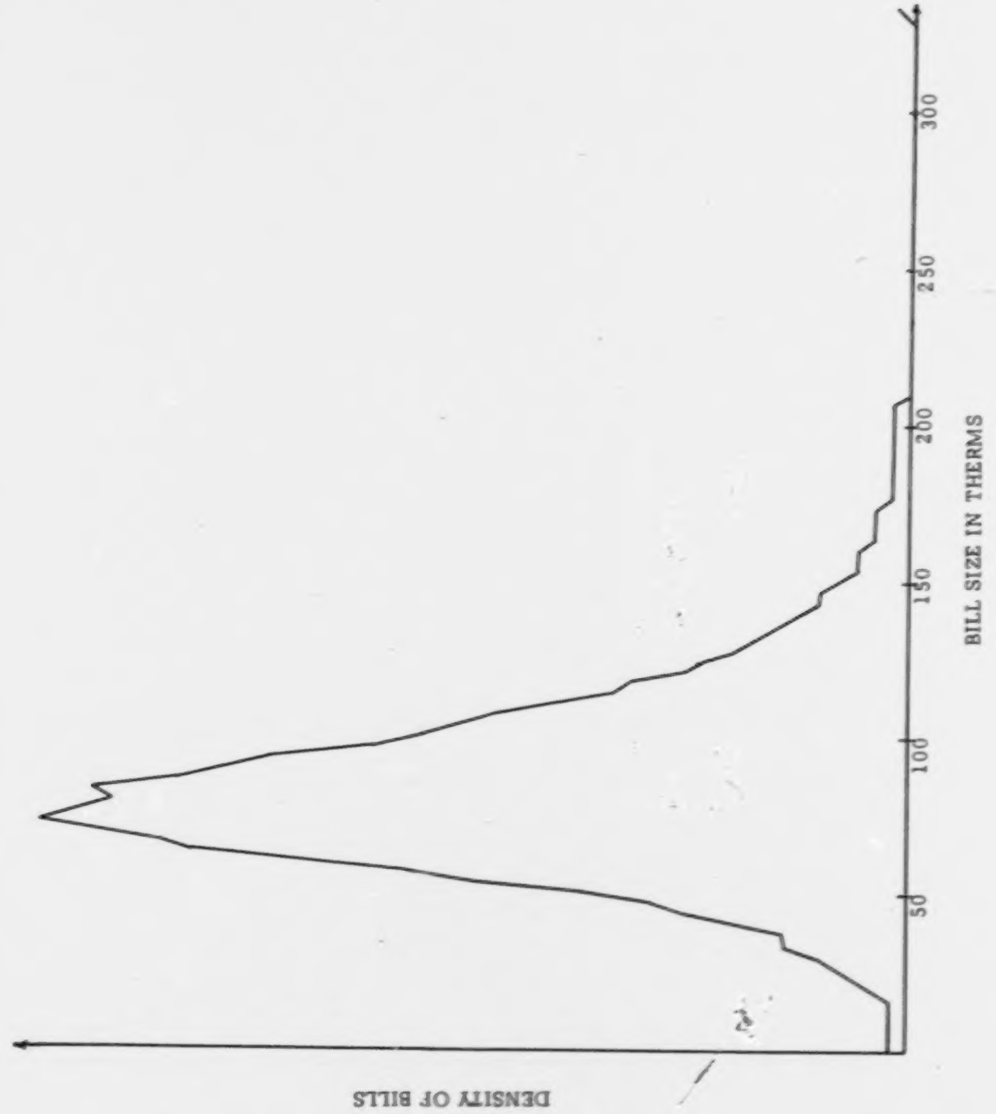


FIGURE 4

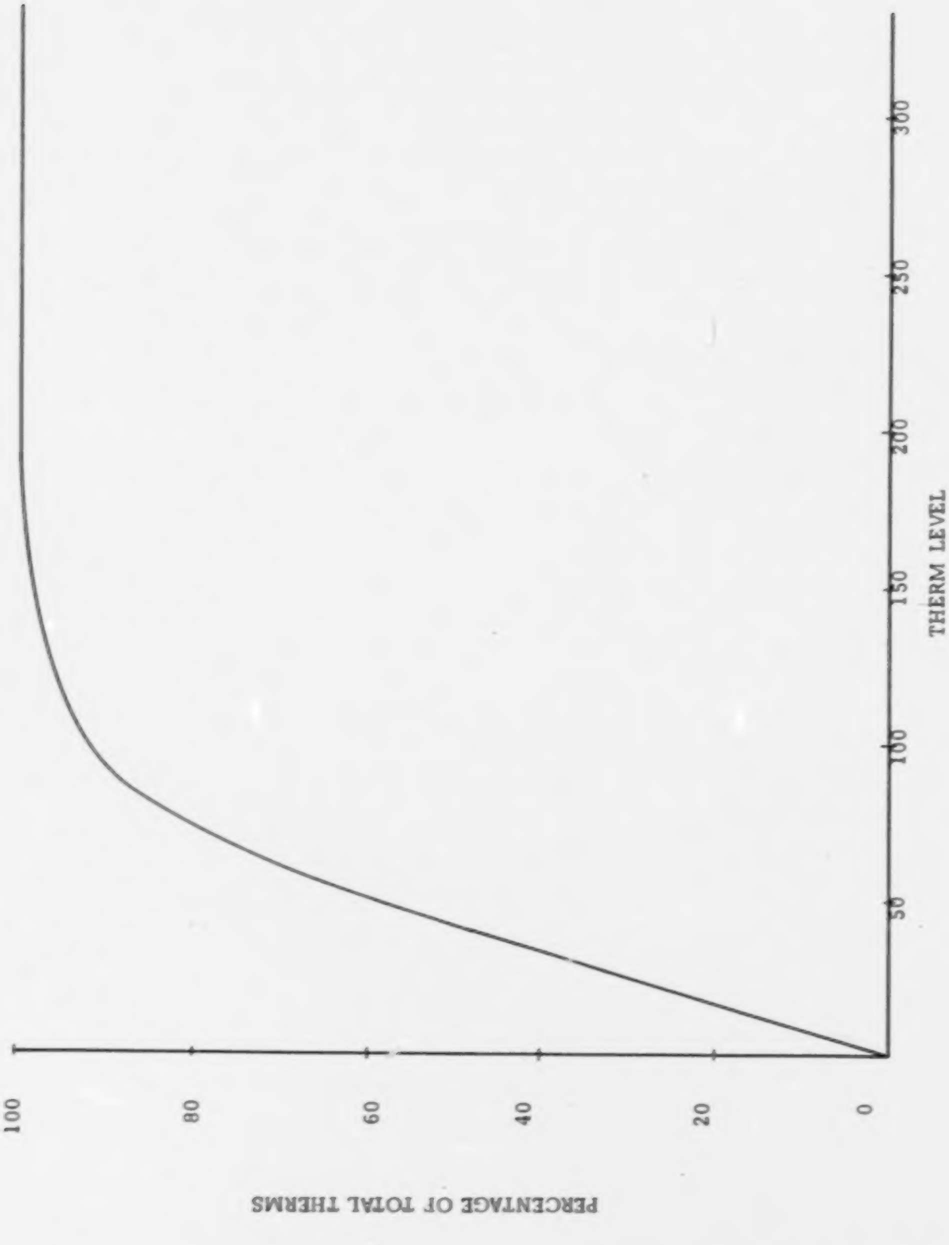

PERCENTAGE OF TOTAL THERMS
100
80
60
40
20
0
50
100
150
200
250
300
THERM LEVEL

FIGURE 5

Reference 1 Ogden, J., "What Happens If - A Planning System for Utilities," *Public Utilities Fortnightly*, March 16, 1972

Reference 2 Liittschwager, J. M., "Mathematical Models for Public Utility Rate Revisions," *Proceedings of the Conference on Public Utility Valuation and Rate Making Process*, Iowa State University, 1969.

Reference 3 Raiffa, H. and Schlaiffer, R., *Applied Statistical Decision Theory*, Harvard University, Boston, Massachusetts, 1961, pp. 225-226.

A SIMULATED INVESTMENT ANALYSIS FOR A GAS PIPELINE COMPANY

Hal Miller

Colorado Interstate Gas Company

Abstract

The supply and demand schedules for gas pipeline companies are probabilistic in form and dynamistic in nature. These factors, along with the other uncertainties associated with gas supply investment decisions, must be considered in order to properly evaluate decision alternatives. These dynamic, uncertain and interrelated decision elements can be properly evaluated through computer based simulation, where each element not known precisely is considered as a random variate, to be simulated. The manifestation of the resulting simulation model is the expected profit and loss (variance from the perfect decision) of each investment alternative, evaluated over its anticipated life.

INTRODUCTION

The populace of the world appears to have an insatiable desire for energy, for as people become more appreciative of what energy can do for them they utilize ever-increasing quantities of it. Each child demands more energy in his lifetime than did his parents and in this quest for an energy-rich Utopia in which he will be free from limitations prescribed by his physical capabilities mankind is creating an energy explosion that is far more staggering than the infamous population explosion. By the turn of the century, less than thirty years time, the world's population is expected to be almost double what it is now, but world's annual consumption of energy is expected to be almost six times the present consumption level. Energy consumption in the United States is expected to be over three times its present level.[1]

---

[1] William T. Reid, "The Melchett Lecture, 1969 - The Energy Explosion," Journal of the Institute of Fuel, February, 1970.

The sheer magnitude of the investment necessary to meet this tremendous growth in demand is going to require a great deal of innovation on the part of energy companies in the formulation of investment strategies. It is going to compel managers to become more cognizant of market reaction to higher prices (which are inevitable) and to more effectively evaluate the risks and uncertainties inherent in these types of investments.

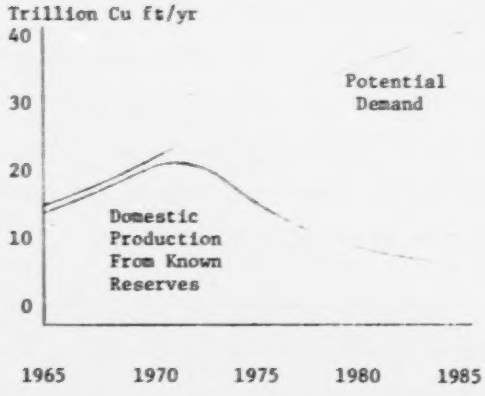
This paper concerns a simulation approach in evaluating energy supply investment strategies. More specifically, it addresses itself to the investment problems currently facing gas pipeline companies.

BACKGROUND

Currently, the natural gas industry is providing about one-third of the energy consumed in the United States. The industry has a current investment in plant and equipment of over $40 billion, or about sixteen percent of the $250 billion currently invested in the U.S. energy industry as a whole.[2] To meet the growing energy demands it has been estimated that $500 billion will be needed to finance investments by energy companies over the next fifteen years.[3] It becomes readily apparent from the graph in Figure 1, which represents an estimation, for the U.S., of the future production capability of the existing gas supply and the potential demand for gas, that the gas industry will probably require a considerable portion of that $500 billion if it is to remain a viable element in the energy industry.

Figure 1

U.S. GAS SUPPLY AND DEMAND



Source: Humble Oil and Refining Company

In the past, gas companies have not been greatly concerned with the uncertainties associated with supply related investments, because most of the risk was carried by gas exploration and production companies and the pipeline company simply made its investment after a gas supply was discovered. Likewise, these companies have not been overly concerned with market reaction to price, because the price of gas has historically been low relative to competing fuels and the increase in price resulting from a particular investment would not have harmed gas' competitive position in the market place. The traditional method used by such a company in formulating supply related investment plans generally assumed that, given no production constraint, the company would continue to maintain its market share. A comparison of the future production capability of the company's present facilities to the demand resulting from the continued market share would provide a forecast of the company's future supply- demand gap, and a determination would, then, be made of the investment necessary to fill this gap.

---

[2] "Statistics of Privately Owned Electric Utilities in the United States," Federal Power Commission, Washington, D.C., December 1971. "1971 Gas Facts," American Gas Association, Arlington, Va., 1971. "Annual Financial Analysis of the Petroleum Industry," The Chase Manhattan Bank, New York, N.Y., August, 1971.

[3] Hollis Dole, Assistant Secretary of Interior, during speech to Financial Analysts Federation, New York, N. Y., May 1972.

Because pipeline companies are most often regulated by some form of governmental agency, they are protected (by either dictate or economics) from invasion by another gas company into their geographic market area. Also, their rates are set to provide for a specific return on investment. Considering only this, it appears that a gas company would have few reservations about making whatever investment deemed necessary to fill the gap mentioned above.

This statement may have been appropriate a few years ago, but the environment acting on this type of decision has become so complex as to completely negate its validity at the present time. The statement neglects two fundamental points. One is that a pipeline company's market, although protected from another gas company, is not protected from invasion from another energy form. Another is that, if the investment were not a prudent one, the regulatory agencies would not allow full return on that investment. These two factors are interrelated, for if a company made an investment to fill the projected supply-demand gap and it turned out that gap had narrowed because of a negative shift in the market share, the company would find that it had over-invested. The consequences, in theory at least, would be a dilution in the company's overall rate of return.

DISCUSSION

Uncertainty is pervasive in the environment of this decision, for investment decisions of a pipeline company are, by nature, very long in term. The ramifications of such a decision can hardly be known precisely. Also, the energy situation is changing very rapidly from both technological and consumption standpoints and this dynamism further augments the uncertainty.

In the analytical approach to this problem the first place uncertainty arises is in the market place. Demand cannot be estimated on a deterministic basis and probabilistic confidence limits should be used to envelop any demand forecast. Directly related to these market uncertainties is the financial risk that the regulators will not allow a return on an "over-investment." The financial risks, however, are not limited to that of the market place or the rate makers. For example, the new supply environment requires that a portion of a pipeline company's future capital expenditures be channeled into exploration of natural gas (traditionally a high risk investment). In addition, a large part of a typical company's future expenditures will be for nonconventional gas supplies. This includes such things as nuclear stimulated gas reserves and coal or oil gasification plants to produce synthetic gas. The political problems associated with coal and oil gasification present risks for these types of investments.

The manifestation of all of these interrelating elements is that a gas pipeline company has dynamic probabilistic supply and demand schedules and in order to properly evaluate supply related investment alternatives these dynamic probabilities have to be considered.

Theoretical Constructs

The basic decision variable is, of course, investment. It is a discrete variable for there is a limited number of investment alternatives available. Two other variables are considered to be directly dependent upon the investment variable -- supply and price. Supply could be said to be functionally related to the investment parameter through the following expression:

$$(1) \quad S_t = S_o + K_1 (I)$$

where:

S_t = supply for some period (t)

S_o = supply for some period (t) if no additional investment were made

K_1 = constant, dependent upon investment alternative and vary overtime

I = additional investment

Because of the regulated nature of the company with its rates based upon return on investment, price could be said to be functionally related to the investment parameter through the following expression:

$$(2) \quad P_t = \begin{cases} (I_o + I)\, K_2/S_t \ , & S_t \leq D_t \\ (I_o + I)\, K_3/D_t \ , & S_t > D_t \end{cases}$$

where:

P_t = price per unit volume for some period (t)

I_o = initial investment

K_2, K_3 = constants which reflect rate of return and cost of service

D_t = demand for some period (t)

S_t = supply for some period (t)

The demand (D_t) is functionally related to price, which is, in turn, related to investment, as indicated in expression (2). The demand variable can be shown to be functionally related to the other variables as follows:

$$(3) \quad D_t = \begin{cases} S_t - K_4(P_t - P_z) & , P_t > P_z \\ S_t & , P_t = P_z \\ S_t - K_5(P_t - P_z) & , P_t < P_z \end{cases}$$

where:

D_t, S_t and P_t are as before

K_4, K_5 = constants reflecting elasticity of demand

P_z = optimum price where supply and demand are at the equilibrium point on the supply-demand schedule

The prime objective of management in selecting specific investment alternatives is, of course, to maximize profits. It can be intuitively shown that profit, in this instance, is maximized when supply is precisely equal to demand. When supply is less than demand there exists an opportunity loss, for the firm is not realizing the sales volume and the subsequent profit, in the form of return on investment, that it could be realizing. When supply exceeds demand, however, the firm has apparently drifted, in theory at least, into the situation where it has made "imprudent" investments and the regulatory agency will not allow the firm to earn on that "unnecessary" investment. Thus, a real loss occurs, which I term a risk loss. Loss, then, is variance from the perfect decision - when supply and demand are equal. In analyzing an investment alternative both profit and loss have to be considered. The goal would then be to select the alternative that optimizes the combination of expected profits and expected losses.

The profit level of investment alternatives for a pipeline company is the resultant return on investment. An indication of the loss level can be determined through the following function:

$$(4) \quad L = \begin{cases} (S_t - D_t)\ K_6\ , & P_t > P_z \\ 0 & , P_t = P_z \\ (S_t - D_t)\ K_7\ , & P_t < P_z \end{cases}$$

where:

S_t, D_t , P_t , and P_z are as before

K_6, K_7 = constants reflecting unit losses

The expected loss for any particular time (t) can be determined, for each investment alternative, as follows:

$$(5) \quad EL_t = \iint_{-\infty}^{+\infty} L \cdot f(S_t) \cdot f(D_t) \cdot dS_t dD_t$$

A present value determination of the expected loss of an alternative over the life of the investment can be conducted as follows:

$$(6) \quad EL = \sum_{t=1}^{n} \left[EL_t \left(\frac{1}{1+i} \right)^t \right]$$

where:

EL_t is as before

i =annual capital discount factor

n =life of the investment in years

The integral in expression (5) can be evaluated by Monte Carlo methods using a normal random number generator on the distributions of supply and demand.

Because of the real-world dynamism and uncertainty, the model developed in this paper is stochastic in nature and uses simulation techniques to evaluate the system's stochastic properties. The basis for the simulation is that each relevant variable that is being estimated, or for some other reason is not known precisely, is considered to be a random variate. A known, or assumed, probability density function is applied to each of these variables to "simulate" its degree of unknownness (for want of a better word). The model enables the user to utilize both subjectively defined density functions and quantitatively determined functions.

As an example of a subjective function, suppose the value of a particular parameter is estimated to be 100 units and the estimator feels that there is a 50-50 chance the real value will fall within ± 10 units of that estimate (and the associated density function is assumed to be normal). Since the 90-110 unit interval contains half the total probability, the probability of the true value lying above 110 is 25 percent. This means that $\sigma(\mu)$ must have a value such that:

$$(7)\quad P(\mu > 110) = P\left[\mu > \frac{110-100}{\sigma(\mu)}\right] = P\,\frac{10}{\sigma(\mu)} = .25$$

From the normal tables,

$$\frac{10}{\sigma(\mu)} = .67$$

$$\sigma(\mu) = 15 \text{ units}$$

The $\sigma(\mu)$ is the standard error (or deviation) and μ is a random variable analogous to the true value of the parameter. The density function for this particular variable would be normal with a mean of 100 units and a standard deviation of 15 units.

The quantitatively determined density functions are simply determined analytically or empirically. An example of a quantitatively determined density function is that of regression equation where the confidence interval is based on the following:

$$(8)\quad Var(Y) = \hat{\sigma}^2\left[1 + \frac{1}{n} + \frac{(x - \bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}\right]$$

where:

$\hat{\sigma}$ = variance of the regression or the estimate of the variance of the errors of observation

n = number of observations

x = independent variable

x_i = observation of independent variable

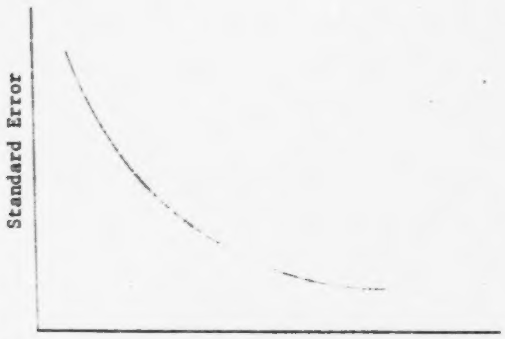
$\bar{x}$ = mean of independent variable

Var Y = variance of the estimated dependent variable

The density function in this case would be normal with the mean being the estimate of Y from the regression equation and the standard deviation being the square root of Var(Y).

In the model, each random variable is simulated through random number generation with each variable's simulation being conducted independently (for those variables whose density functions are independent of each other). The optimum number of simulations has been determined through analysis of the standard error of the estimate. The graph in Figure 2 shows the typical relationship between the error and the number of simulations.

Figure 2



Number of Simulations

As the number of simulations increases the standard error more closely approximates the theoretical value. In most cases 100 simulations proved to be adequate, for any additional incremental shift in standard error could not justify the incremental cost of additional simulation.

General Model Description

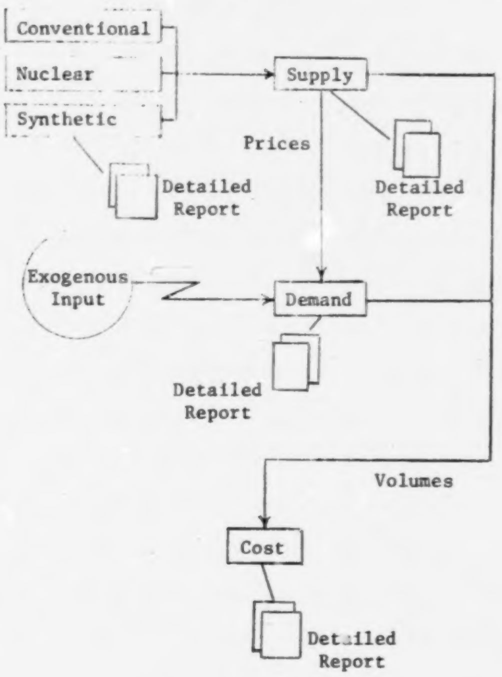
A sub-model has been developed for each major decision variable, such as supply, demand, or expected profit and loss. The desired output from each model is an aggregate forecast of the particular decision element and its standard error of estimate for each forecast period.

The general logic for the system of sub-models is shown on the diagram in Figure 3. Each box can be considered as being a sub-model. There are three sources of data for the supply model -- the conventional, nuclear and synthetic sub-models. The output of these models is the estimated supply parameters resulting from conventional natural gas production, nuclear stimulated production and coal or oil gasification, respectively. Each of these outputs contains volumetric forecasts, price forecasts, and estimated volume and price variances for each year of the forecast horizon. One of the three supply sub-models is computerized and that is the synthetic model, which contains a simulation of coal hydrogenation variables. The other two models are based on analytic estimation of the parameters. The simulated price and its deviation from the supply model is used as input for the demand model, which uses other exogenous factors to simulate demand. The simulated demand and its deviation, and the simulated supply and its deviation are used as input for the cost model which, when provided the investment dollars, return criteria and other relevant data, simulates income and opportunity cost for each year. The present value of those future incomes and opportunity costs are also calculated by this sub-model.[4] The computerized models were developed over a period of time and both FORTRAN and Time Sharing BASIC languages were used.

---

[4] A more detailed discussion of each of the sub-models is exhibited in the Appendix.

Figure 3



RESULTS

This model could be used for various types of comparative investment analyses for a pipeline company, ranging from analysis of a detailed and intricate investment strategy to that of comparing the ramifications of two simple investment alternatives. To illustrate the use of the model three hypothetical investment strategies for a typical pipeline company have been analyzed. The only investment decision that varies from one strategy to the next in this example concerns the type of coal hydrogenation process to be employed. All other decision elements are assumed to remain constant.

At the present time a dozen hydrogenation processes are being promoted by various manufacturing and engineering firms. These processes are all at various stages of development, but none have been used commercially. Each process is unique from a financial viewpoint. For instance, some processes require a lower level of investment than others but, consequently, have higher operating costs. Also, the level of risk is different for each process so that one process might result in a gas price considerably below the others, but it has a smaller chance of becoming a commercial reality in time to do any good.

In view of all these uncertainties and variabilities, the most logical thing to do would be to wait for the processes to become commercially available before making the selection. That would take the guess work out of the decision, but that is not feasible, for the promoters of these processes are demanding development capital and, in order to receive the benefit of early use of one of these plants, commitments have to be made now. These commitments are extremely large, for the cost of building one of these plants is on the order of $250-$300 million.

The three types of processes presented in this paper are termed A, B and C. Process A is closest to being commercially available in that a portion of the process has been in use for quite some time. There is the smallest risk associated with this process, but is has a low efficiency so its estimated gas price will be greater than the others. It is anticipated that this process will be the first to be available for use - possibly by 1977. It is estimated that process B will result in the cheapest gas price and will require less investment than the others. At the same time, however, it will probably have higher operating costs. It is estimated that this process could be used by late 1978. The program to develop this process is of a crash type. Consequently, the risks associated with it are greater than for the others. Process C won't be available until about 1982 and its estimated resultant price will be somewhere between the other two processes. This process is being developed over a longer period of time so the risks associated with it are less than with process B.

The first illustrative hypothetical alternative is that of investing in one plant of each of the processes and bringing them on stream at their earliest possible dates. The A plant is to be on stream in 1977, the B plant in 1978 and the C plant in 1982. The second investment alternative is that of investing in one A plant and two B plants. The A plant is to be on stream in 1977 and the two B plants in 1982. The third alternative is to invest in a B plant to be on stream in 1978, and a C plant to be on stream in 1982.

The results of the investment alternatives are shown on Table 1 through Table 3.[5] For the first alternative, the present value of the income (return) and opportunity costs are $451 million and $81 million, respectively. For the second alternative, those values are $411 million and $76 million, respectively, and for the third alternative, they are $414 million and $49 million, respectively. Alternative one maximizes return and alternative three minimizes the opportunity costs. The annual income or return figures, shown on the tables should also be of interest for these values can be used to approximate the net investment of an alternative for any particular point in time. For instance, the 1991 net investment for the three alternative strategies is about $1,050 million, $850 million, and $950 million, respectively.[6]

---

[5] All figures are hypothetical

[6] These values are calculated by dividing the annual income figures by 8-1/2 percent, which is the assumed hypothetical rate of return.

Table 1
INVESTMENT ALTERNATIVE I

	Supply		Demand		Income	Cost
Year	BCF	Std Dev	BCF	Std Dev	$Million	
1972	443	17	426	13	17.21	0.71
1973	463	23	426	16	17.95	1.73
1974	476	27	448	17	19.75	1.25
1975	494	32	473	17	21.74	0.93
1976	511	37	476	20	24.64	1.56
1977	556	42	467	25	35.00	6.78
1978	586	48	470	25	42.98	10.64
1979	550	53	431	24	42.69	11.28
1980	567	60	409	22	42.33	16.42
1981	535	67	485	26	52.07	4.59
1982	584	73	447	29	62.99	18.03
1983	617	81	437	24	73.75	29.80
1984	590	90	458	20	79.60	22.93
1985	580	100	447	23	80.41	21.74
1986	562	112	472	26	86.00	13.55
1987	543	122	517	24	91.60	7.08
1988	527	134	574	33	100.69	4.16
1989	508	144	493	39	89.88	8.53
1990	505	158	481	29	87.90	9.64
1991	517	171	490	40	89.59	10.70

PRESENT VALUE INCOME	PRESENT VALUE COST
$ 451.33	$ 81.32

Table 2
INVESTMENT ALTERNATIVE II

	Supply		Demand		Income	Cost
Year	BCF	Std Dev	BCF	Std Dev	$Million	
1972	443	17	426	13	17.21	0.71
1973	463	23	426	16	17.95	1.73
1974	476	27	448	17	19.75	1.25
1975	494	32	473	17	21.74	0.93
1976	511	37	476	20	24.64	1.56
1977	556	42	466	25	35.00	6.78
1978	586	48	470	25	42.98	10.64
1979	550	53	431	24	42.69	11.28
1980	567	60	409	22	42.36	16.42
1981	535	67	485	26	52.07	4.59
1982	634	75	444	30	59.15	24.72
1983	617	85	441	27	60.73	23.39
1984	599	96	471	25	66.45	16.57
1985	580	107	451	29	65.88	16.68
1986	562	120	476	32	70.00	10.52
1987	543	131	529	29	74.86	5.40
1988	527	142	601	42	82.60	4.80
1989	508	154	499	49	72.22	7.58
1990	505	167	485	34	87.38	9.92
1991	517	181	493	48	72.25	9.24

PRESENT VALUE INCOME	PRESENT VALUE COST
$ 411.64	$ 75.82

Table 3
INVESTMENT ALTERNATIVE III

	Supply		Demand		Income	Cost
Year	BCF	Std Dev	BCF	Std Dev	$Million	
1972	443	17	426	13	17.21	0.71
1973	463	23	426	16	17.95	1.73
1974	476	27	448	17	19.75	1.25
1975	494	32	473	17	21.74	0.93
1976	511	37	476	20	24.64	1.56
1977	526	42	475	24	26.85	2.98
1978	566	48	484	25	35.96	5.35
1979	520	53	449	25	36.22	5.25
1980	537	60	431	24	37.08	8.77
1981	505	67	511	30	43.81	1.22
1982	564	73	469	31	58.25	9.24
1983	587	81	461	29	71.43	18.71
1984	569	89	481	25	75.93	12.22
1985	560	100	457	27	75.03	13.09
1986	532	112	479	29	78.72	8.15
1987	513	122	527	26	83.24	4.46
1988	497	133	590	37	89.98	6.27
1989	478	144	490	43	80.41	7.21
1990	505	158	485	31	88.41	9.35
1991	487	171	495	43	80.61	8.45

PRESENT VALUE INCOME	PRESENT VALUE COST
$ 414.21	$ 49.36

The objective has to include the present values of both income and opportunity costs. The income figure represents the expected payoff or monetary value of the particular strategy employed and the cost figure represents the risk of monetary loss that could occur through employment of that strategy. Members of Management who are responsible for selecting the investment strategy have to weigh the profit potential against the risk. All the model can do is provide data for consideration by Management. The important ingredient in this decision process which now becomes prevalent is the relative degree of risk aversion of the members of Management. This factor is combined with the simulation results to form the utility of each investment alternative and it is hoped that through this process of combining the quantitative and qualitative elements of the decision-making process, the result is the best solution to the problem.

REFERENCES

American Gas Association, "1971 Gas Facts," Arlington, Va., 1971

Bierman, Harold, Jr., Bonini, Charles P., et al., "Quantitative Analysis for Business Decisions," Homewood, Illinois, Richard D. Irwin, Inc., 1965

Chase Manhattan Bank, "Annual Financial Analysis of the Petroleum Industry," New York, New York, August, 1971.

Federal Power Commission, "Statistics of Privately Owned Electric Utilities in the United States," Washington, D. C., December, 1971.

Frederick, Donald G., "Industrial Pricing Decision Using Bayesian Multivariate Analysis," Journal of Marketing Research, May, 1971.

Reid, William T., "The Melchett Lecture, 1969 - The Energy Explosion," Journal of the Institute of Fuel, February, 1970.

Schlaifer, Robert, "Analysis of Decisions Under Uncertainty," New York, New York, McGraw-Hill, 1967.

Schlaifer, Robert, "Probability and Statistics for Business Decisions," McGraw-Hill, New York, New York, 1959.

APPENDIX

Conventional Supply - An estimation of a typical company's supply of conventional natural gas can be determined analytically through statistical analysis with the use of Gompertz curvefitting techniques. Using this technique, the density function of the conventional natural gas supply is determined through the regression confidence interval calculation exhibited in the DISCUSSION portion of the paper. The density function can be revised by the company's supply personnel to reflect their subjective estimation of the potential supply to be discovered and their evaluation of the company's ability to compete for that supply. The result of this analysis is estimated supply, prices, return and respective variances for each year in the forecast. The prices are the wholesale prices necessary to provide a specified return on the investment required to market the gas. The return indicates that proportion of the price which is necessary to provide the required return.

Nuclear Supply - The supply, prices, return, and respective variances for nuclear stimulated gas production are estimated on a subjective basis with consideration being given to various political and technological problems which might be encountered. The prices and return are as discussed in the Conventional Supply section, above. The subjective density function associated with this "sub-model" is defined according to the procedure outline in the paper.

Coal Supply - Prices, return, and their respective variances for coal hydrogenated gas is determined through a computerized simulation model. This model considers the level of investment required for a particular hydrogenation process and calculates a gas price that would be required to earn a specified rate of return.[1] The variables that are being estimated (and therefore treated as random variables) are:

[1] depreciation
[2] interest rates
[3] debt/equity ratio
[4] tax rates
[5] labor costs
[6] material costs
[7] equity return
[8] coal costs

The output of this model includes a detailed breakdown of various factors relevant to an economic analysis of a hydrogenation plant, a sample of which is shown in Table 1. The general logic of the simulation aspect of this model is similar to that of the demand model which is discussed later.

The supply of hydrogenated gas and its related variance is subjectively estimated, with consideration being given to the reliability of the technology involved and the degree to which that technology has been proved through actual application. One of the biggest factors in this consideration is the variability associated with the estimation of the earliest "on stream" date for one of these plants.

TABLE 1
COST OF SERVICE (MANUFACTURED COST OF GAS)
(AMOUNTS IN THOUSANDS OF DOLLARS)

Particulars	1973
Total Facilities Investment	$ 76,898
Working Capital	5,383
Total Capital Investment	$ 82,281
Cost of Service	
Operation and Maintenance	
Direct Labor	1,669
Maintenance	2,568
Supplies	385
Administrative and General	
Supervision	167
Payroll	184
General	2,395
Insurance	369
Coal Cost	17,082
Water Cost	147
Other Direct Materials	811
Depreciation	3,845
Return	7,558
Federal Income Tax	4,001
State Income Tax	288
Other Local Tax	1,153
Subtotal	$ 42,622
Contingencies	767
Subtotal	$ 43,389
Byproduct Credit (Char, Sulfur, Power)	11,454
Total Cost of Service/Year	$ 31,935
Cost of Gas Determination	
Annual Gas Production, MMBTU	51,903
Basic Gas Prices, Cents/MMBTU	61.53
Royalty Cost, Cents/MMBTU	2.50
Final Gas Price, Cents/MMBTU	64.03
Mean Value of Basic Gas Price	61.50
Standard Deviation of Gas Price	2.20

---

[1] It might be well to point out that the same type of simulation model could be developed for the conventional and nuclear supply cases. It is felt at this time, however, that the subjective aspects of these analyses negate the need for such sophistication.

Total Supply Model - The total supply schedule is estimated through a computerized simulation model which takes, as input, all of the supply, price and variance estimates of the previously mentioned sub-models and independently generates random numbers representing each of these variables. Since retail prices are required to determine market demand and the prices determined thus far are wholesale prices, an estimation is needed for the retail mark-up in each consumption category. This in itself is a considerable task for utility pricing strategies vary substantially from one sales category to another. In addition, an estimation is needed of the retailers' revenue requirements. To account for the uncertainties involved in this procedure, these mark-ups are also simulated. The results from this model are used as input for two other models. The retail prices for each consumption category and their respective standard errors are transmitted to the demand model and the aggregate total supply and its standard error are transmitted to the cost model. An example of the information generated by this model is shown in Table 2.

Demand Model - The total demand schedule is estimated through a computerized econometric and simulation model which takes, as input, the retail prices of each sales category generated in the total supply model discussed above and their respective standard errors. Other exogenous factors used in this model include population, per capita income, competing energy prices and household formations. As in all the models discussed thus far, each estimated variable is treated as a random variable and is simulated through random number generation. These estimated variables, which are treated as random statistics, include both exogenous and endogenous variables. In addition, a confidence interval has been established for each of the fifteen regression equations in the model. The regression equations represent market share elasticity or sensitivity to price, income, or various other factors. These interval calculations have been based on the equation exhibited in the Theoretical Constructs section of the paper. Random numbers analogous to each dependent variable are generated, based on the confidence intervals. These random numbers can be thought of as simulating the degree to which the regression equation does not statistically explain the behavior of that dependent variable.

A flow diagram of the basic logic of this model is shown on Figure 1. The output of the model includes a breakdown of various factors relevant to a market analysis, an example of which is shown in Table 3. The total demand and its standard error are transmitted to the cost model for further use.

Cost Model - The term cost model is somewhat misleading for the model's function is to summarize the results of all the previous models and simulate the financial ramifications of the entire process. The model takes, as input, the total

TABLE 2

Year	SUPPLY		CITY GATE		RESIDENTIAL		COMMERCIAL		INDUSTRIAL		ELECTRIC GEN	
	BCF	STD DEV	PRICE	STD DEV	PRICE	STD DEV	PRICE	STD DEV	PRICE	STD DEV	PRICE	STD DEV
1972	443	17	0.27	0.03	0.73	0.06	0.61	0.05	0.30	0.03	0.30	0.03
1973	463	23	0.28	0.05	0.76	0.08	0.64	0.07	0.31	0.05	0.31	0.05
1974	476	27	0.29	0.07	0.80	0.11	0.67	0.09	0.33	0.07	0.36	0.07
1975	494	32	0.31	0.08	0.83	0.13	0.69	0.11	0.34	0.08	0.34	0.08
1976	511	37	0.34	0.09	0.89	0.14	0.75	0.13	0.38	0.09	0.38	0.09
1977	556	42	0.43	0.10	1.00	0.16	0.85	0.14	0.47	0.10	0.47	0.11
1978	586	48	0.52	0.12	1.11	0.19	0.95	0.16	0.56	0.12	0.56	0.12
1979	580	53	0.56	0.13	1.17	0.20	1.01	0.18	0.60	0.13	0.60	0.13
1980	567	60	0.60	0.15	1.24	0.22	1.07	0.19	0.64	0.15	0.64	0.15
1981	536	67	0.64	0.16	1.30	0.24	1.13	0.21	0.68	0.16	0.68	0.16
1982	584	73	0.74	0.18	1.43	0.26	1.25	0.23	0.78	0.18	0.78	0.18
1983	617	81	0.82	0.20	1.54	0.28	1.35	0.25	0.87	0.20	0.87	0.20
1984	590	90	0.85	0.21	1.60	0.31	1.41	0.27	0.90	0.21	0.90	0.21
1985	500	100	0.86	0.23	1.66	0.35	1.46	0.29	0.93	0.23	0.93	0.23
1986	562	112	0.91	0.26	1.72	0.35	1.51	0.31	0.96	0.26	0.96	0.26
1987	543	122	0.95	0.28	1.79	0.38	1.57	0.34	1.01	0.28	1.01	0.28
1988	527	136	0.98	0.31	1.85	0.40	1.63	0.36	1.04	0.31	1.04	0.31
1989	508	147	1.02	0.35	1.93	0.44	1.69	0.40	1.08	0.35	1.06	0.35
1990	505	158	1.04	0.38	1.98	0.48	1.74	0.43	1.10	0.38	1.10	0.38
1991	517	171	1.05	0.42	2.03	0.51	1.78	0.47	1.12	0.42	1.12	0.42

supply estimate and its standard error, the total demand estimate and its standard error, and the various prices, returns and their respective standard errors, and simulates the resultant income and opportunity cost. Income, in this instance, is before taxes and financial costs.[2] Opportunity costs are basically measures of variance from the perfect decision and, as mentioned before, arise in two instances - when demand exceeds supply, in which case profit opportunities are not fully utilized, and when supply exceeds demand, in which case an overinvestment has been made and the utility regulators will disallow a return on that "imprudent" investment. These income and opportunity cost estimates for each year of the forecast are discounted to the present to arrive at a present value figure so that comparative analyses can be conducted for the various investment alternatives. The general logic in this model is exhibited on Figure 2 and a sample output is exhibited as Tables 1-3 in the Results section of the paper.

---

[2] This is the manner in which return on investment is calculated for a utility.

Figure 1

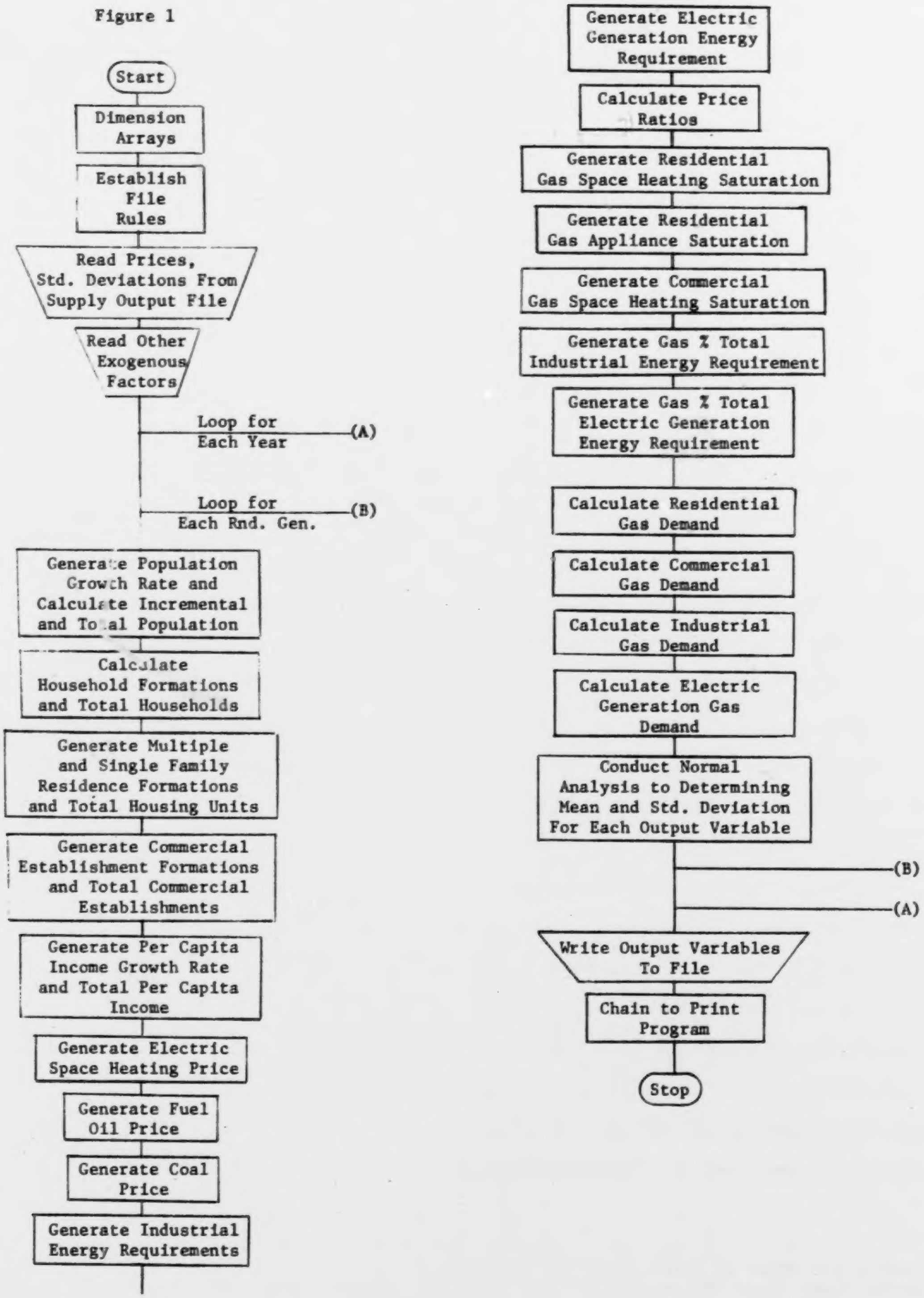

Start
Dimension Arrays
Establish File Rules
Read Prices, Std. Deviations From Supply Output File
Read Other Exogenous Factors
Loop for Each Year
(A)
Loop for Each Rnd. Gen.
(B)
Generate Population Growth Rate and Calculate Incremental and Total Population
Calculate Household Formations and Total Households
Generate Multiple and Single Family Residence Formations and Total Housing Units
Generate Commercial Establishment Formations and Total Commercial Establishments
Generate Per Capita Income Growth Rate and Total Per Capita Income
Generate Electric Space Heating Price
Generate Fuel Oil Price
Generate Coal Price
Generate Industrial Energy Requirements
Generate Electric Generation Energy Requirement
Calculate Price Ratios
Generate Residential Gas Space Heating Saturation
Generate Residential Gas Appliance Saturation
Generate Commercial Gas Space Heating Saturation
Generate Gas % Total Industrial Energy Requirement
Generate Gas % Total Electric Generation Energy Requirement
Calculate Residential Gas Demand
Calculate Commercial Gas Demand
Calculate Industrial Gas Demand
Calculate Electric Generation Gas Demand
Conduct Normal Analysis to Determining Mean and Std. Deviation For Each Output Variable
(B)
(A)
Write Output Variables To File
Chain to Print Program
Stop

TABLE 3

NATURAL GAS MARKETS
TYPICAL GAS COMPANY

1972

Population	2,285,370
Number Households	724,128
Single Residences	543,320
Commercial Establishments	71,298
Per Capita Income	$ 3,578.69

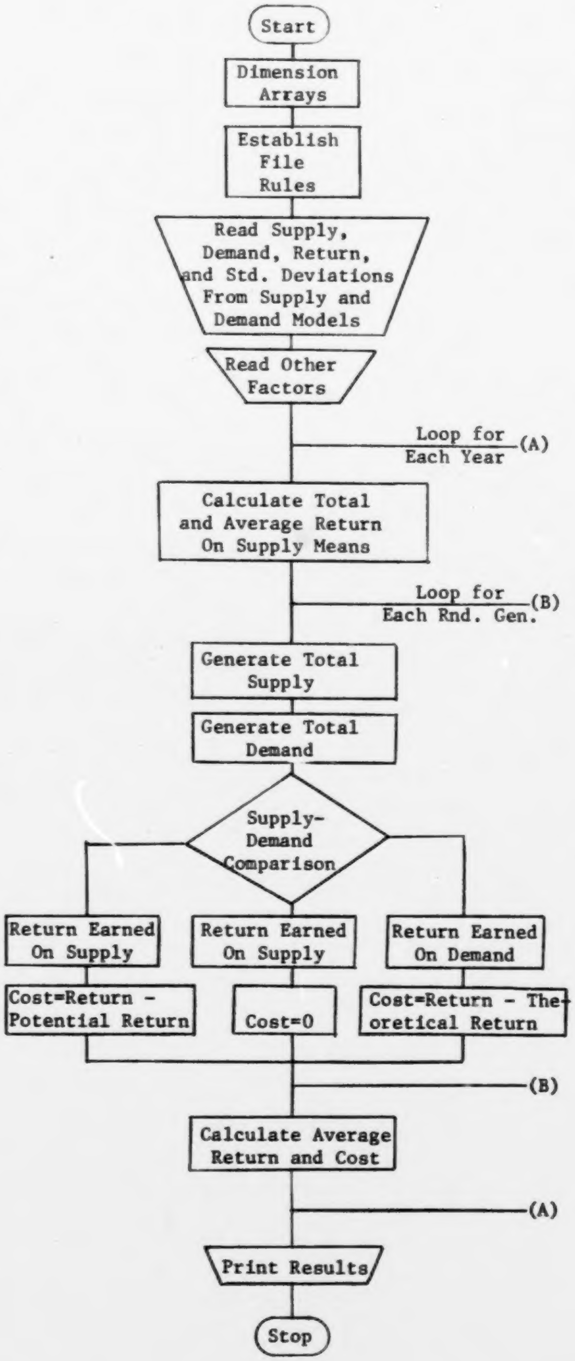
Gas Price	Per MMBtu
Residential	$ .735
Commercial	.615
Industrial	.301
Electric Generation	.301
Electric Price	4.903
Residential Fuel Oil Price	1.079
Commercial Fuel Oil Price	1.079
Industrial Fuel Oil Price	.549
Electric Generation Fuel Oil Price	.457
Industrial Coal Price	.421
Electric Generation Coal Price	.312

Residential Saturations	
Central Heating	95.0 %
Ranges	38.6 %
Water Heaters	90.0 %
Clothes Dryers	11.6 %
Commercial Saturation	95.0 %
Industrial Percentage	64.0 %
Electric Generation Percentage	46.1 %

GAS CONSUMPTION REQUIREMENTS

	MMCF	STD. DEV.
Residential	80,378.0	1,085.0
Commercial	50,266.4	2,230.2
Industrial	86,732.5	6,748.0
Electric Generation	52,324.1	8,060.8
Total	269,701.0	12,484.9
Other Sales	134,250.0	-
Lost & Unaccounted For	22,217.3	686.7
Total Requirements	426,168.0	13,171.5

Figure 2


Start
Dimension Arrays
Establish File Rules
Read Supply, Demand, Return, and Std. Deviations From Supply and Demand Models
Read Other Factors
Loop for Each Year (A)
Calculate Total and Average Return On Supply Means
Loop for Each Rnd. Gen. (B)
Generate Total Supply
Generate Total Demand
Supply-Demand Comparison
Return Earned On Supply
Return Earned On Supply
Return Earned On Demand
Cost=Return - Potential Return
Cost=0
Cost=Return - Theoretical Return
(B)
Calculate Average Return and Cost
(A)
Print Results
Stop

Session 14: Languages for Simulation
Chairman: Philip Kiviat, Federal ADP Simulation Center

Simulation languages are past adolescence and nearing maturity. This is seen in the two papers in this session that describe improved and extended versions of the widely-used simulation languages GPSS and GASP, and by the papers that bring linguistics and mathematics to bear in providing more useful and usable simulation tools.

Papers

"An Interactive Simulation Programming System Which Converses in English"
George E. Heidorn, Naval Postgraduate School

"GASP IV: A Combined Continuous/Discrete FORTRAN Based Simulation Language"
Nicholas R. Hurst and A. Alan B. Pritsker, Purdue University

"NGPSS/6000: A New Implementation of GPSS"
Lee Rogin, Naval Air Development Center and Karen Ast,
Jerry Katzke, Jim Nickerson, Julian Reitman,
Norden - A Division of United Aircraft Corp.

"CMS/1 - A Corporate Modeling System"
R. F. Zant, Clemson University

Discussants

Paul F. Wyman, Pennsylvania State University
Arnold Ockene, Securities Industries Automation Corporation
Robert H. Downs, Systems Control, Inc.

AN INTERACTIVE SIMULATION PROGRAMMING SYSTEM WHICH CONVERSES IN ENGLISH

George E. Heidorn

Naval Postgraduate School

Abstract

In this paper an experimental system for producing simulation programs through natural language dialogue with a computer is described. With the current version of this system, which operates under CP/CMS on an IBM 360/67, a queuing problem may be stated to the computer in English. The system checks the problem description for completeness, and if necessary, asks questions which may be answered in English. Then it can produce an English description of the problem as it "sees" it and a GPSS program for performing the simulation. The user may then modify the problem through further dialogue to produce additional programs, as desired. A complete sample problem is included in the paper.

1. Introduction

This paper reports on work which is being done at the Naval Postgraduate School to develop a system for performing simulation analyses through natural language (e.g. English) interaction with a computer. The eventual goal is to enable an analyst seated at a terminal to "talk" with the computer about his simulation problem in his own natural language, and have the computer "understand" the problem and do the simulation, reporting the results in the same natural language. In such a system the computer would function as an intelligent assistant with great computing powers - actually a combination programmer and computer.

The status of this system as of the time of this writing is that it can carry on a dialogue in English about a simple queuing problem and then produce a GPSS program to do the simulation. This is essentially the programmer part of the total system described above.

This project has been influenced by work being done in the areas of simulation programming languages, artificial intelligence, and linguistics. The entity-attribute-value "world view" of both SIMSCRIPT [1] and GPSS [2] plays a central role in this system. Ideas have been borrowed from artificial intelligence work on natural-language question-answering systems [3,4], and the language processing has been done within the framework of Stratificational Linguistics [5,6].

This paper is intended to serve as an introduction to the system being developed and to demonstrate some of the capabilities of the current version. Another introductory paper based on a slightly earlier version is available [7], and a very detailed technical report has been prepared [8]. This paper consists primarily of a sample problem and an explanation of its overall processing, without details of the linguistic aspects.

2. The Sample Problem

Figures 1, 2, and 3 show a terminal session with this system for a sample problem. The statement numbers which appear there were added for ease of reference in the discussion. All of the lower case typing was done by the user, and all of the UPPER CASE by the computer.

```
 1.  vehicles arrive at a station.

 2.  the station has just one pump.

 3.  a vehicle will leave the station immediately
     after arriving if the length of the line at
     the pump is not less than two.

 4.  otherwise, it is serviced there; then it
     leaves.

 5.  service times are exponential, with a mean of
     5 minutes for cars and 9 minutes for trucks.

 6.  three quarters of the vehicles are cars and
     one fourth of them are trucks.

 7.  is the statement of the problem complete?

     HOW OFTEN DO THE VEHICLES ARRIVE AT THE
     STATION?

 8.  arrivals are normally distributed with a mean
     of 8 minutes and a standard deviation of 2
     minutes.

     HOW LONG SHALL THE SIMULATION BE RUN?

 9.  the basic time unit to be used in the model
     is 30 seconds.

     HOW LONG SHALL THE SIMULATION BE RUN?

10.  eight hours.

             THE PROBLEM STATEMENT IS COMPLETE.
```

Figure 1. The English statement of the problem being given to the system.

The first figure shows the problem description being given to the system. As can be seen there, the user entered six statements and then asked if the problem description was complete. The system realized that information about the inter-arrival times was missing, so it asked a question about that, which the user responded to with a complete sentence. Then the system asked how long the model was to be run, but the user chose to give information about the basic time unit instead, which would have been the subject of the next question. After processing the reply,

11. describe the problem in english.

```
     THE VEHICLES ARRIVE AT THE STATION.  THE
TIME BETWEEN ARRIVALS OF THE VEHICLES AT THE
STATION IS NORMALLY DISTRIBUTED, WITH A MEAN OF
8 MINUTES AND A STANDARD DEVIATION OF 2 MINUTES.
75 PERCENT OF THE VEHICLES ARE CARS, AND THE
REST ARE TRUCKS.  AFTER ARRIVING AT THE STATION,
IF THE LENGTH OF THE LINE AT THE PUMP IN THE
STATION IS LESS THAN 2, THE VEHICLE WILL BE
SERVICED AT THE PUMP IN THE STATION.  OTHERWISE,
THE VEHICLE WILL LEAVE THE STATION.  THE TIME
FOR THE VEHICLES TO BE SERVICED AT THE PUMP IN
THE STATION IS EXPONENTIALLY DISTRIBUTED, WITH A
MEAN OF 5 MINUTES FOR THE CARS, AND 9 MINUTES
FOR THE TRUCKS.  AFTER BEING SERVICED AT THE
PUMP IN THE STATION, THE VEHICLES LEAVE THE
STATION.
     THE SIMULATION IS TO BE RUN FOR 8 HOURS,
USING A BASIC TIME UNIT OF 30 SECONDS.
```

Figure 2. The English problem description produced by the system.

the system again asked the question about the run time. This time the user responded with the phrase, "eight hours". Then the system announced that it had a complete problem statement, i.e. one for which it could produce a GPSS program.

In the second figure there is a description of the problem "in the computer's own words", which the user can read carefully to see if the system "understood" him. Then the third figure shows the GPSS program produced by the system for performing the simulation.

3. The Overall Approach

If a simulation programmer were given a queuing problem stated in a natural language, he would probably read it one or more times to form a "mental image" of the system being described and to note the points of interest in it. If the description were not clear to him or if essential information were missing, he might ask questions of the writer until he felt that he completely understood the problem and had all the information he needed to do the program. At this point

12. write a gpss program for this problem.

```
       SIMULATE
       RMULT      277,423,715,121,655,531,999,813
 STAT1 EQU        1,F,Q
 PUMP2 EQU        2,F,Q
 CAR2  EQU        2,T
 2     TABLE      M1,1,1,2
 TRUC3 EQU        3,T
 3     TABLE      M1,1,1,2
 1     FUNCTION   RN1,C24
0.0,0.0/.100,.104/.200,.222/.300,.355/
.400,.509/.500,.690/.600,.915/.700,1.200/
.750,1.390/.800,1.600/.840,1.830/.880,2.120/
.900,2.300/.920,2.520/.940,2.810/.950,2.990/
.960,3.200/.970,3.500/.980,3.900/.990,4.600/
.995,5.300/.998,6.200/.999,7.000/1.000,8.000/
 2     FUNCTION   RN2,C29
0.0,-3.000/.012,-2.250/.027,-1.930/.043,-1.720/
.062,-1.540/.084,-1.380/.104,-1.260/.131,-1.120/
.159,-1.00/.187,-.890/.230,-.740/.267,-.620/
.334,-.430/.432,-.170/.500,0.0/.568,.170/
.666,.430/.732,.620/.770,.740/.813,.890/
.841,1.000/.869,1.120/.896,1.260/.916,1.380/
.938,1.540/.957,1.720/.973,1.930/.988,2.250/
1.000,3.000/
 3     FUNCTION   P1,D2
CAR2,10/TRUC3,18/
 4     FUNCTION   RN3,D2
.750,CAR2/1.000,TRUC3/
 1     FVARIABLE  16+4*FN2
*
*      THE VEHICLES ARRIVE AT THE STATION.
       GENERATE   V1
       ASSIGN     1,FN4
       TEST L     Q$PUMP2,2,ACT2
       TRANSFER   ,ACT3
*
*      THE VEHICLES LEAVE THE STATION.
 ACT2  TABULATE   P1
       TERMINATE
*
*      THE VEHICLES ARE SERVICED AT THE PUMP.
 ACT3  QUEUE      PUMP2
       SEIZE      PUMP2
       DEPART     PUMP2
       ADVANCE    FN3,FN1
       RELEASE    PUMP2
       TRANSFER   ,ACT2
*
*      TIMING LOOP
       GENERATE   960
       TERMINATE  1
       START      1
       END
```

Figure 3. The GPSS program produced by the system.

he might state the problem "in his own words" to the writer as a check on his understanding of it. Finally, he would think about the problem in terms of the concepts of the computer language he planned to use, and then he would write the program.

The computer system being described here serves the same role as the simulation programmer described above. Therefore, it was designed to follow essentially the same overall procedure as he does, as can be seen from the example in Figures 1 through 3. The computer's counterpart of the programmer's mental image, the Internal Problem Description, is central to the operation of this system and will be discussed first, followed by discussions of the English input, the English output, and the GPSS program.

4. The Internal Problem Description

The Internal Problem Description (IPD) is an entity-attribute-value data structure for holding information about a particular problem in a language-independent form. Entity-attribute-value data structures have been widely used both in artificial intelligence applications and in simulation programming systems such as SIMSCRIPT and GPSS. In the IPD an entity is represented by a "record", which is just a list of attribute-value pairs. Some of the records in an IPD represent physical entities, such as a car or a dock, and others represent abstract entities, such as an action or a function. The attributes which a record has depend, of course, upon the entity being represented. The value of an attribute may be simply a number or a name, or it may be a pointer to another record.

A queuing problem typically deals with physical entities, such as cars or ships, moving through a system to be serviced in some manner at other physical entities, such as a pump or a dock, in the system. Here, the former of these are termed "mobile entities", and the latter are called "stationary entities". (In SIMSCRIPT these are temporary and permanent entities, and in GPSS they are transactions and facilities and storages.) As the mobile entities move through the system, they engage in "actions" at the stationary entities. Some of these actions are instantaneous, such as arrive and leave, and are called "events"; others, such as service and load, consume time and are referred to as "activities" here.

The IPD describes the flow of mobile entities through a system, by specifying the actions which take place there and their interrelationships. Each of these actions is represented by a record which has attributes to furnish such information as the type of action, the entity doing the action, the one to whom the action is being done, the location where it happens, how long it takes, how often it occurs, and what happens next.

A graphic portrayal of the IPD built by the system for the sample problem of Figure 1 appears in Figure 4. In the figure each record of the IPD is represented by a box, with the name of the record appearing at the top of the box. With the exception of MEMORY and 'ACTNLIST', these names do not actually exist within the computer, but were

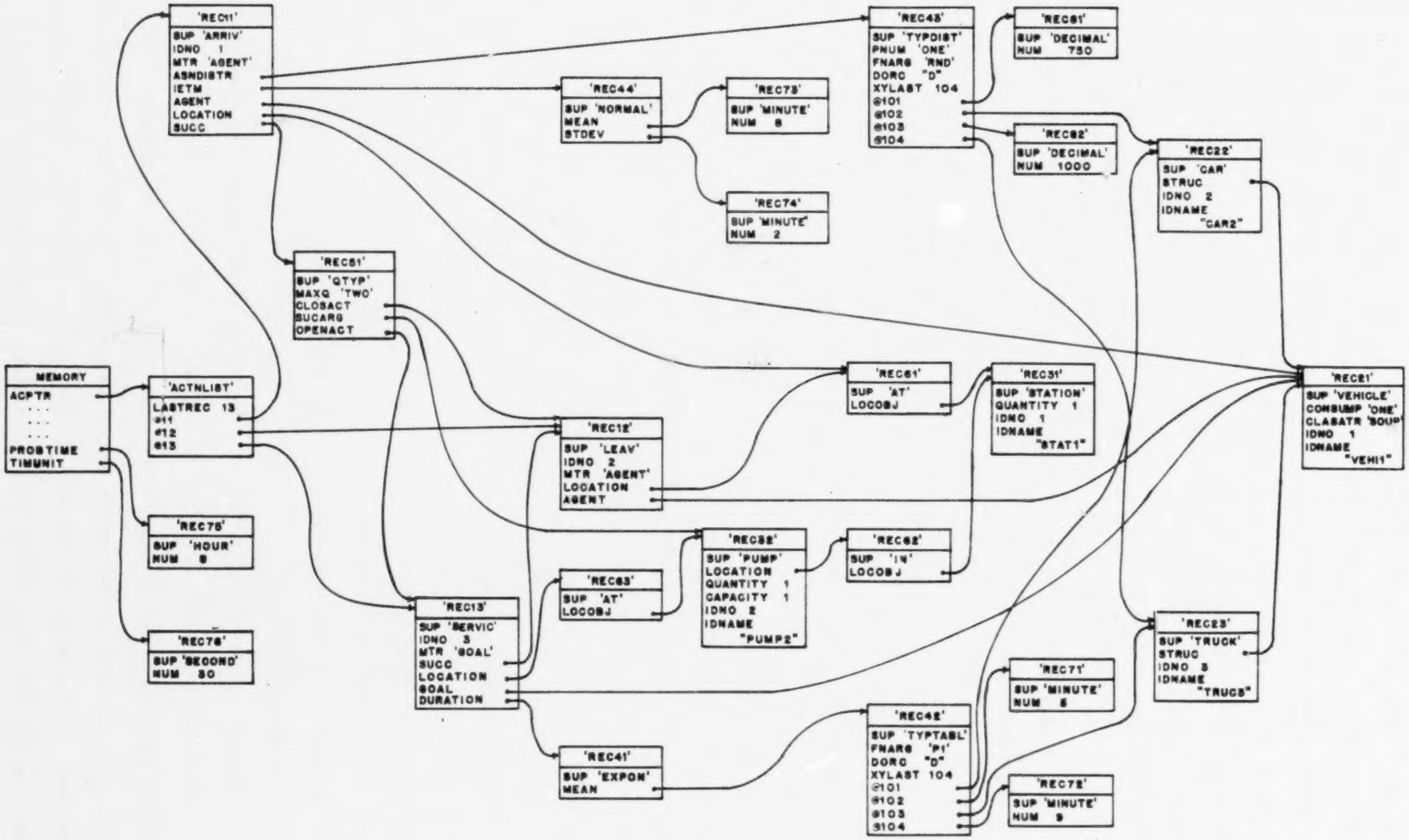

MEMORY
ACPTR
PROBTIME
TIMUNIT
'ACTNLIST'
LASTREC 13
@11
@12
@13
'REC75'
SUP 'HOUR'
NUM 8
'REC76'
SUP 'SECOND'
NUM 30
'REC11'
SUP 'ARRIV'
IDNO 1
MTR 'AGENT'
ASNDISTR
IETM
AGENT
LOCATION
SUCC
'REC51'
SUP 'QTYP'
MAXQ 'TWO'
CLOSACT
SUCARG
OPENACT
'REC44'
SUP 'NORMAL'
MEAN
STDEV
'REC73'
SUP 'MINUTE'
NUM 8
'REC74'
SUP 'MINUTE'
NUM 2
'REC12'
SUP 'LEAV'
IDNO 2
MTR 'AGENT'
LOCATION
AGENT
'REC13'
SUP 'SERVIC'
IDNO 3
MTR 'GOAL'
SUCC
LOCATION
GOAL
DURATION
'REC63'
SUP 'AT'
LOCOBJ
'REC41'
SUP 'EXPON'
MEAN
'REC32'
SUP 'PUMP'
LOCATION
QUANTITY 1
CAPACITY 1
IDNO 2
IDNAME
"PUMP2"
'REC62'
SUP 'IN'
LOCOBJ
'REC61'
SUP 'AT'
LOCOBJ
'REC31'
SUP 'STATION'
QUANTITY 1
IDNO 1
IDNAME
"STAT1"
'REC43'
SUP 'TYPDIST'
PNUM 'ONE'
FNARG 'RND'
DORC "D"
XYLAST 104
@101
@102
@103
@104
'REC81'
SUP 'DECIMAL'
NUM 750
'REC82'
SUP 'DECIMAL'
NUM 1000
'REC42'
SUP 'TYPTABL'
FNARG 'P1'
DORC "D"
XYLAST 104
@101
@102
@103
@104
'REC71'
SUP 'MINUTE'
NUM 5
'REC72'
SUP 'MINUTE'
NUM 9
'REC22'
SUP 'CAR'
STRUC
IDNO 2
IDNAME
"CAR2"
'REC23'
SUP 'TRUCK'
STRUC
IDNO 3
IDNAME
"TRUC3"
'REC21'
SUP 'VEHICLE'
CONSUMP 'ONE'
CLASATR 'SOUP'
IDNO 1
IDNAME
"VEHI1"

Figure 4. The Internal Problem Description (IPD)

placed on the drawing simply to furnish a means of referring to the various records in the discussion which follows. In each box the attribute-value pairs of the record are shown, with the attribute name or number on the left and its value on the right. Many of the values are pointers to other records in the IPD, in which case an appropriate arrow is drawn.

It can be seen that MEMORY is the only record which is not pointed to by some other record. It plays a rather central role in the IPD, being used both to hold global information about the problem (e.g. problem time and the basic time unit) and to serve as sort of a directory into the rest of the IPD. Only one portion of the "directory" was included in this drawing in order to keep the number of lines at a minimum. The portion included is the "action list" ('ACTNLIST'), which, as can be seen, contains pointers to each of the three action records. Not included in the drawing are the lists for mobile entities ('MOBLIST'), stationary entities ('STALIST'), distributions ('DSTRLIST'), and successor descriptors ('SCSRLIST'). The action list may be considered to be the most important list, because of the key role which actions play in a problem description.

Every IPD record, except for the MEMORY record and the lists just mentioned, has a SUPerset attribute pointing to the "named record" representing the concept of which this record is a specific instance. For example, the SUP attribute of the first action record (REC11) points to the named record 'ARRIV', indicating that this action (vehicles arrive at a station) is a specific instance of the concept "arriv". Named records contain information about the words and concepts for queuing problems and are entered into the system at initialization time.

Each action record in the IPD must have either an AGENT or a GOAL which points to a mobile entity record. The AGENT of an action is the one doing the action, and the GOAL is the one to whom the action is being done. The MTR attribute tells which of these two is pointing to a mobile entity. Each action record must also have a LOCATION attribute pointing to a "location descriptor" record, which in turn points to a stationary entity record. An event like 'ARRIV' or 'ENTER' must have an IETM (inter-event time) attribute to specify the time between occurrences of the event, and an activity (e.g. 'SERVIC' or 'LOAD') must have a DURATION attribute to specify the time taken to perform the activity. These times can be constants, standard probability distributions, functions, or combinations of these, some of which can be seen in the drawing. REC42 in the drawing is a function which has the records for car and truck as its X values and the records for 5 minutes and 9 minutes as its Y values. The ASNDISTR attribute of an 'ARRIV' specifies the percentages of the various kinds of entities which arrive, in the form of a cumulative probability distribution. REC43 in the drawing furnishes an example of this. (The NUM attribute of a 'DECIMAL' record is considered to be in parts-per-thousand.) The attributes DORC, FNARG, and PNUM which appear in REC42 and REC43 are needed for encoding the GPSS program.

Each action record, except a 'LEAV', must have a SUCCessor attribute to specify which action the mobile entity of this action is involved in next. The value of SUCC may simply be a pointer to another action record, or it may be a pointer to a "successor descriptor" record. REC51 in the drawing is an example of one of the five types of successor descriptors currently available in the system. This particular record, which is a 'QTYP' can be interpreted as saying, "If the length of the line at the pump (SUCARG) is less than two (MAXQ), go to be serviced (OPENACT); otherwise, leave (CLOSACT)." The other types of successor descriptors available handle such situations as "If the pump is busy, the vehicle leaves.","Cars are serviced, and trucks leave.", and "Half of the vehicles are serviced, and the rest leave."

It can be seen in the drawing that the records for 'CAR' and 'TRUCK' each have a STRUCture attribute pointing to the record for 'VEHICLE'. This is related to the idea of the "assignment distribution" (ASNDISTR), and essentially means that cars and trucks may be referred to as vehicles in the problem description. The attribute CLASATR (class attribute) in the 'VEHICLE' record indicates what is the distinguishing attribute of any records which have a STRUC attribute pointing to this record. (SOUP is synonymous with SUP in this case.) Part of the usefulness of the STRUC attribute is that it avoids some unnecessary duplication of information. For example, the value of the CONSUMPtion attribute is the same for both cars and trucks in this problem, so it need be stored only once, up in the 'VEHICLE' record. (CONSUMP indicates how many units of a resource are required by a mobile entity.)

Each entity and action record is assigned an identification number (IDNO) for use in the GPSS program. The value of IDNAME is formed by concatenating the first three or four letters of the NAME of the SUP of a record with the value of its IDNO. CAPACITY, QUANTITY, and CONSUMP are given default values of 1 or 'ONE' if they are not specified in the original problem description. ('ONE' is a named record, with a SUP of 'UNIT' and a NUM of 1.)

5. The English Statement of the Problem

At the beginning of a terminal session there is no IPD. It is the English input that furnishes the information which enables the system to build one. For example, when sentence 1 was processed, REC21 and REC31 were created and their SUP and IDNO attributes were given values. Also, REC61 was created, with values for its SUP and LOCOBJ attributes, and REC11 was created, with values for its SUP, IDNO, AGENT, and LOCATION. When sentence 3 was processed, REC12, REC63, and REC51 were created with some attributes, and the SUCC attribute was added to REC11.

Sentences which occur in natural language descriptions of queuing problems can be considered to fall into two categories: "action sentences" and "attribute sentences". An action sentence has as its main verb an action verb, which is modified by phrases and clauses to specify the values of the attributes of the action. For example,

> After arriving, if the dock is available, the ship is unloaded at the dock.

is an action sentence; the action is "unload", its goal is "ship", its location is "dock", its predecessor is "arrive", and its condition is "dock available". It should be noted that the order of most of the phrases and clauses in this sentence could be changed without altering the information content.

An attribute sentence has as its main verb an attribute verb, and is used to specify the value of some attribute of some record in the IPD. For example,

> The time to unload the ship is 8 hours.

says that the value of the "time" (actually duration) attribute of the action record "unload ship" is "8 hours". An equivalent statement would be

> It takes 8 hours to unload the ship.

The English statement of the problem must describe the flow of mobile entities through the queuing system. This is done by saying something about each action that takes place there, and how it is related to other actions. Each mobile entity must "arrive" at or "enter" the system. Then it may go through one or more other actions, such as "service," "load," "unload," and "wait." Then, typically, it "leaves" the system. The order in which these actions take place must be made explicit by the use of subordinate clauses beginning with such conjunctions as "after," "when," and "before," or by using the adverb "then". If the order of the actions depends on the state of the system being simulated, an "if" clause may be used to specify the condition for performing an action. Then a sentence with an "otherwise" in it would be used to give an alternate action to be performed when the condition is not met.

In the sample problem in Figure 1, sentences 1, 3, and 4 are action sentences describing the flow of vehicles through the station. This same information could be given in a wide variety of ways. For example, the following would be acceptable as input:

> Arrivals of vehicles occur at a station. If the length of the line at the pump in the station is less than 2 when a vehicle arrives, it will be serviced at the pump. Otherwise, it will leave immediately. After being serviced, a vehicle leaves the station.

In addition to describing the flow of mobile entities through the queuing system, the English statement of the problem must also furnish other information needed to simulate the system, such as the various times involved. It is necessary to specify the time between arrivals, the time required to perform each activity, the length of the simulation run, and the basic time unit to be used in the GPSS program. Also, the quantity of each stationary entity should be specified. (A quantity of one is assumed otherwise.) Other information, such as that of sentence 6 in the example, may also be given.

In the sample problem, sentences 2, 5, 6, 8 and 9 are attribute sentences furnishing this additional information. Just as with the action sentences, this information could be given in a wide variety of ways. For example, sentence 2, which specifies the values of both the location and quantity attributes of the pump, could be stated in at least the following three ways:

> There is one pump in the station.
>
> The quantity of pumps in the station is one.
>
> There is one pump, and the pump is in the station.

Each attribute sentence is essentially of the form, "attribute of entity equals value." The name of the attribute may be given explicitly, as is "quantity" in the second sentence above, or it may be implied by the verb or the type of value or some other characteristic of the sentence. For example, the verb "hold" implies the capacity attribute in the following sentence:

> The station can hold three cars.

An equivalent statement would be:

> The capacity of the station is 3 cars.

In each of the following, the attribute is implied by the type of value:

> The pump is in the station.
>
> The pump is green.

These are equivalent to:

> The pump is located in the station.
>
> The color of the pump is green.

In sentence 8 of the sample problem, the attribute "inter-event time" is implied.

The entity referred to may be a physical entity, such as pump and station in the above examples, or it may be an abstract entity, such as an action or a probability distribution. When it is an action, the infinitive or present participle form of the action verb, along with appropriate modifiers, is usually used to identify the action. For example, the following four sentences are all equivalent:

> The time to service a vehicle at the pump is 5 minutes.
>
> The time for servicing a vehicle at the pump is 5 minutes.
>
> The time for a vehicle's servicing at the pump is 5 minutes.
>
> The servicing of a vehicle at the pump takes 5 minutes.

Any one of the first three underlined phrases could have appeared in the fourth sentence of this example too.

The value part of an attribute sentence can take many different forms, as can be seen in the sample problem and in the above examples. Especially important in simulation models are quantitative values, of which there are several forms. The following phrases are examples of quantitative values:

> ten tons
>
> from 10 to 20 minutes
>
> 9 minutes for trucks and 5 minutes for cars
>
> exponentially distributed with a mean of two hours

In the last phrase above there is actually a second level of attribute and value, i.e. the mean (attribute) of the exponential distribution (entity) is two hours (value). This can also be seen in sentences 5 and 8 of the sample problem.

Whenever some action is referenced in either an action sentence or an attribute sentence, only enough modifiers to distinguish that action from others have to be used. For example, sentence 8 of the sample problem could have begun, "Arrivals of vehicles" or "Arrivals of vehicles at the station." However, it was sufficient to simply say, "Arrivals," because only one "arrive" action had been previously mentioned in the problem description.

Some use of pronominal reference is allowed, also. For instance, "it" is considered to refer to the most recent non-person mobile entity or stationary entity mentioned, whichever would make a meaningful sentence in the queuing problem context. Similarly, "there" is considered to be a substitute for the most recent location phrase. Both of these can be seen in sentence 4 of the sample problem, where "it" refers to the "vehicle" and "there" means "at the pump." The following sentence would have exactly the same meaning as sentence 4:

> Otherwise, it is serviced at it;
> then it leaves.

In this case the middle "it" would be taken as "the pump," because a location phrase requires a stationary entity in the queuing problem context.

Most sentences of the input text are completely parsed (at least implicitly), i.e. every word and phrase must be accounted for. However, the system is also capable of extracting meaning from some sentences just by the appearance of certain "keywords." For instance, if the words "time" and "unit" and some time phrase (e.g. "30 seconds") appear in a sentence, the time phrase is considered to be the value of the TIMUNIT attribute of MEMORY. Similarly, the appearance of "GPSS" or "program" results in the GPSS program being produced. In the sample problem, sentences 7, 9, 11, and 12 are keyword sentences.

In the English input the user may either state the complete problem immediately, or he may state just some part of it and then let the system ask questions to obtain the rest of the information, as was done in Figure 1. Each time the system asks a question, it is trying to obtain the value of some one essential attribute. A question may be answered by a complete sentence (e.g. statement 8) or simply by an appropriate phrase (e.g. statement 10) to furnish a value for the attribute, or the question may be ignored and a sentence with some other information given (e.g. statement 9).

6. The English Problem Description Produced by the System

The overall manner in which the English problem description is produced by the system can be seen by comparing the information in the text of Figure 2 with the information in the IPD of Figure 4. The first paragraph is produced by going down the action list and saying something about the attributes of each action. The very first action is simply stated with an action sentence containing information about the type of action, its AGENT and/or GOAL, and its LOCATION. If the IETM or DURATION attribute has a simple value, it will be included also, as a prepositional phrase (e.g. "every 8 minutes" or "for 5 minutes"). Otherwise, a separate statement in the form of an attribute sentence will be made about the IETM or DURATION, as can be seen in the figure. If the action has an ASNDISTR, a statement will then be made about it, as also can be seen in the figure. Finally, a statement of the form "After ..., ..." is produced from the SUCC attribute. The exact form of this statement depends upon the type of value which SUCC has. It can be seen in the figure that a 'QTYP' successor descriptor actually results in two sentences, with the first one having an "if" clause and the second one beginning with "otherwise".

When describing an action which has already been mentioned in a successor statement, it is not necessary to produce a simple sentence about that action. If the action has a non-simple DURATION and/or a SUCC, the appropriate statements about these can immediately be made. This is the case for the 'SERVIC' action in the example. No output was produced from the 'LEAV' action, because it had already been mentioned in a successor statement and it had no additional attributes to be described.

If a stationary entity has a QUANTITY or CAPACITY attribute with a value greater than 1. a statement will be made about it shortly after the entity is first mentioned in an action sentence (e.g. "There are 2 pumps in the station." or "The capacity of the station is 8 vehicles."). After describing the actions and the entities, a separate one-sentence paragraph is produced with the values of PROBTIME and TIMUNIT of MEMORY, as can be seen in the figure.

7. The GPSS Program Produced by the System

The manner of producing the GPSS program is similar to that for the English problem description, but it involves going down several lists, not just the action list. As was mentioned earlier, these other lists are not shown in the IPD drawing in Figure 4. Their contents will be given in parentheses at appropriate points in the following discussion, however.

The first bit of GPSS program produced is a standard SIMULATE card and RMULT card. Then a pass is made down the stationary entity list (REC31, REC32) to produce an EQU card for each stationary entity, to relate its IDNAME and its IDNO and to define it as a facility or a storage and a queue. If either the QUANTITY or CAPACITY attribute is greater than 1, an appropriate STORAGE definition card is also produced. Then a similar pass is made down the mobile entity list (REC21, REC22, REC23) to output an EQU card and a TABLE card for each type of mobile entity that will actually appear in the simulation (i.e. those records that do not have a CLASATR attribute). In the example, nothing is included for 'VEHICLE' because any vehicle that appears is either a car or a truck. The tables defined will be used to record transit times during the simulation.

Next, a standard FUNCTION 1 for the exponential distribution and a standard FUNCTION 2 for the unit normal distribution are produced if they are required by the problem. Then a pass is made down the distribution list (REC41, REC42, REC43, REC44) to define a FUNCTION for each record that requires one. In the example, FUNCTION 3 comes from REC42, and FUNCTION 4 comes from REC43. This is followed by a similar pass down the successor descriptor list (REC51) to define a FUNCTION for each record that requires one. This pass produced nothing in the example.

Then the records in the distribution list are looked at once again to define an FVARIABLE for each normal distribution used in the problem. One of these appears in the example. The numbers 16 and 4 appear there for the mean and standard deviation rather than 8 and 2, as might be expected, because the basic time unit to be used for this problem was specified as 30 seconds rather than 1 minute. The number of each FUNCTION and FVARIABLE defined in the above passes is stored as the IDNO attribute of the record which caused the definition, for use in later processing.

After the definitions have been taken care of, a pass is made down the action list to produce the executable blocks which describe the flow of transactions through the program (which corresponds to the flow of mobile entities through the actual system). For each action a blank comment card (with an asterisk in column 1), followed by a comment card with a simple action sentence on it is immediately put out. This is then followed by the blocks appropriate to this action.

The group of blocks produced from an action actually has two parts, the first of which depends upon the type of action and the second of which depends upon the type of value the SUCC attribute has. For example, an 'ARRIV' usually produces a GENERATE and an ASSIGN, a 'LEAV' produces a TABULATE and a TERMINATE, and most activities produce a sequence like QUEUE, SEIZE, DEPART, ADVANCE, and RELEASE, or minor variations thereof. A 'QTYP' successor descriptor results in a TEST, followed by a TRANSFER (if necessary), and a simple SUCC results in an unconditional TRANSFER, as can be seen in the example. If the 'LEAV' and 'SERVIC' actions had been in reverse order in the action list, the resulting GPSS program would not have needed the two unconditional TRANSFER's which appear in this program, and they would have been suppressed.

The contents of most of the argument fields of the various blocks depend, of course, upon the attributes of the records in the IPD. For example, argument A of the GENERATE block is V1 here because FVARIABLE 1 corresponds to the normal distribution which is the value of the IETM attribute of the 'ARRIV' action. Similarly, argument B of the ASSIGN block (which assigns the transaction type, either 2 or 3, to parameter 1 of the transaction) comes from the ASNDISTR attribute of the same action. Arguments A, B, and C of the TEST block and argument B of the TRANSFER come directly from the attributes SUCARG, MAXQ, CLOSACT, and OPENACT of the 'QTYP' record. The LOCATION attribute determines the A argument for such blocks as QUEUE, DEPART, SEIZE, and RELEASE, as can be seen in the example.

It can also be seen that argument A of the ADVANCE block (the mean advance time) references FUNCTION 3, which was defined from the 'TYPTABL' record which specifies the mean of the DURATION of the 'SERVIC' action. When a transaction enters that ADVANCE block, the appropriate mean time will be obtained from FUNCTION 3 using the value of parameter 1 which was ASSIGN'ed to it when it "arrived". This will then be modified by a value from FUNCTION 1 to yield a service time from the desired exponential distribution. The B argument of the last TRANSFER gets its value directly from the SUCC attribute of the 'SERVIC' action. All actions are referenced by names of the form "ACTi", where i is the value of the action's IDNO attribute.

Finally, after the blocks for the actions are put out, a standard "timing loop" is produced to govern the run length of the simulation. The value in the A argument of the GENERATE block comes from PROBTIME of MEMORY. In the example this value is 960, because there are 960 30-second periods in 8 hours.

8. The System

The computer system developed for this project is in the form of a 5000-statement FORTRAN program called NLP (Natural Language Processor), which is intended to be useful for a wide range of natural-language, man-machine communication tasks. When run under the CP/CMS time-sharing system on an IBM 360/67, it requires a virtual machine with 350K bytes of storage. The program consists of about 100 routines, ranging in size from one which simply unpacks a four-byte word to another which is a compiler for a grammar-rule language. One large group of routines provides list-processing capabilities. The main routine serves as a monitor to provide for interaction with the user.

Before NLP can process a queuing problem, it must be initialized with information about the relevant words and concepts and about the grammars of the languages to be used (currently English and GPSS) and how text is to be processed for these languages. Information about words and concepts is entered by means of "named record" definitions, and the grammars and processing are specified by "decoding rules" and "encoding rules". Each of these is discussed in detail in Reference 8.

9. Computer Time

There are at least three different kinds of time which can be reported for a job run on a time sharing system. The "virtual CPU time" does not include system overhead and is essentially the time that the job would take if run under a batch system. The "total CPU time" includes system overhead, most of which is for paging, and depends somewhat on the current load on the system. "Elapsed time" is the time that the user spends sitting at the terminal and can be very highly dependent on the current load.

For the sample problem the virtual CPU time was 77 seconds for Figure 1, 26 seconds for Figure 2, and 45 seconds for Figure 3, for a total of 148 seconds. The total CPU time was 156 seconds for Figure 1, 41 seconds for Figure 2, and 65 seconds for Figure 3, for a total of 262 seconds. The elapsed time for this problem may vary from one half hour to two hours, depending on the load on the system. Due to improvements in the program, the times given here average about 35 percent less than those given in Reference 7.

10. Conclusion

The system described here is considered to be in its early stages of development. It is already quite capable, as can be seen from the sample problem, but it certainly is not yet ready for production use and may not be for at least a few years.

In line with the overall goal stated in the Introduction, work is currently being done on developing the capability for having the system perform the simulation, rather than just producing a GPSS program. This will make it possible to give the user more control over the actual simulation and also make it possible to report the results in the language of the problem. It is intended that facilities for aiding statistical analyses will be incorporated, too. Along with this, work is continually being done to expand both the kinds of problems that the system can handle and the language which it will accept.

References

1. Markowitz, H. M., Hausner, B., and Karr, H.W. SIMSCRIPT A Simulation Programming Language. Prentice-Hall, Englewood Cliffs, N.J., 1963.

2. International Business Machines. General Purpose Simulation System/360 - Introductory User's Manual. Publication H20-0304, Data Processing Division, White Plans, N.Y., 1967.

3. Minsky, M. L. (Ed.), Semantic Information Processing. The M.I.T. Press, Cambridge, Mass., 1968.

4. Simmons, R. F. Natural language question-answering systems: 1969. Comm. ACM 13, 1 (January 1970), 15-30.

5. Lamb, S. M. Outline of Stratificational Grammar, Georgetown University Press, Washington, D. C., 1966.

6. Lockwood, D. G. Introduction to Stratificational Linguistics, Harcourt Brace Jovanovich, Inc., New York, 1972.

7. Heidorn, G. E. Natural Language inputs to a simulation programming system - an introduction. Technical report NPS-55HD71121A, Naval Postgraduate School, Monterey, Calif., December 1971.

8. Heidorn, G. E. Natural language inputs to a simulation programming system. Technical report NPS-55HD72101A, Naval Postgraduate School, Monterey, Calif., October 1972.

Acknowledgements

The author would like to express his appreciation to Richard C. Hansen, Robert T. McGee, Eldon S. Baker, Frederick H. Hemphill, Robert J. Williams, Alfred H. Mossler, and John H. Rickelman, former students at the Naval Postgraduate School, for their contributions to this research. This work was partially supported by the Information Systems Program of the Office of Naval Research as Project NR 049-314, and the facilities of the W. R. Church Computer Center were utilized.

Keywords

simulation programming, GPSS, natural language, artificial intelligence

GASP IV: A COMBINED CONTINUOUS/DISCRETE, FORTRAN BASED SIMULATION LANGUAGE

by

Nicholas R. Hurst
A. Alan B. Pritsker
Center for Large-Scale Systems
Purdue University

Abstract

GASP IV is an extension of the next event simulation language GASP II. A generalization of the definition of "event" and additions to the language structure enable GASP IV to be used for continuous or combined models while retaining the full power of GASP II for discrete models.

Continuous system state description may be in the form of a set of algebraic and/or differential equations. GASP IV handles the details of state and event control (including state variable integration when necessary), information storage and retrieval, system performance data collection and analysis, and report and plot generation.

In addition to the models which can be coded in GASP II, the following types of models have been successfully coded in GASP IV: Systems Dynamics Models (Industrial, Urban, and World Dynamics Models); Mechanical Impact Models; and Chemical Process Models.

In each case, an analyst familiar with GASP II has been able to quickly write the GASP IV code.

Introduction

GASP IV is a new simulation language with new capabilities. Although it is an extension of GASP II, it provides many of the capabilities normally associated with continuous simulation languages. These additional capabilities are integrated into the GASP II structure resulting in a conceptually and physically integrated language. Because GASP II is well documented (References 3 and 4), this paper will emphasize those features and capabilities of GASP IV not included in GASP II.

GASP IV consists of a set of FORTRAN subroutines organized to assist the analyst in preparing discrete, contiuous, or combined simulation models. GASP IV formalizes an approach to the preparation of such models by providing an appropriate world-view supported by prepared subroutines which handle the problem-independent structure of the model. The world-view provided describes the status of the subject system in terms of a set of state variables and a set of entities with their associated attributes. The GASP IV simulation philosophy is that the dynamic simulation of a system can be obtained by modeling the events of the system and advancing time from one event to the next. This philosophy presumes an expanded definition of "event" which will be stated later.

Every GASP IV simulation model consists of:

1) A set of subroutines which describe a system's operating rules. (Subroutines defining events, conditions causing events, and the trajectories of the state variables.)
2) Lists and matrices which store information.
3) An executive routine.

The set of subroutines describing the operating rules represent the technological logic of the system being studied. The lists and matrices represent the specific entities, their attributes, and associated control information. Variables common to many simulation programs are defined and provided as GASP variables requiring the user to define only problem dependent, non-GASP, variables.

The executive routine and its supporting subroutines provide the nine functions shown below:

1) State and event control including state variable integration when necessary.

    SUBROUTINE GASP

2) System initialization.

    SUBROUTINE DATIN

    SUBROUTINE CLEAR

    SUBROUTINE SET

3) Information storage and retrieval.
    - SUBROUTINE FILEM (IFILE)
    - SUBROUTINE RMOVE (NTRY, IFILE)
    - SUBROUTINE CANCL (NTRY)
    - SUBROUTINE COPY (NTRY)

4) Location of specified state conditions.
    - FUNCTION KROSS (IKRSG, IDRSD, CONST, LDIR, TOL)
    - SUBROUTINE FIND (XVAL, MCODE, IFILE, JATT, NTRY, TOL)

5) System performance data collection.
    - SUBROUTINE COLCT (XX, ICLCT)
    - SUBROUTINE TMST (XX,T,ISTAT)
    - SUBROUTINE HISTO (XX, A,W,IHIST)
    - SUBROUTINE GPLOT (IPLOT,ITAPE, NVARP, LCODE, TIME, P)

6) Statistical computation and reporting.
    - SUBROUTINE PRNTQ (IFILE)
    - SUBROUTINE PRNTS
    - SUBROUTINE SUMRY

7) Monitoring and error reporting.
    - SUBROUTINE MONTR
    - SUBROUTINE ERROR (KODE)

8) Random variate generation.
    - FUNCTION DRAND (ISTRM)
    - FUNCTION UNFRM (ULO,UHI,ISTRM)
    - FUNCTION RNORM (IPAR,ISTRM)
    - FUNCTION RLOGN (IPAR, ISTRM)
    - FUNCTION ERLNG (IPAR,ISTRM)
    - FUNCTION GAMA (IPAR,ISTRM)
    - FUNCTION BETA (IPAR,ISTRM)
    - FUNCTION NPOSN (IPAR,ISTRM)
    - FUNCTION GAM (AK,ISTRM)

9) Miscellaneous support.
    - FUNCTION SUMQ (JATT,IFILE)
    - FUNCTION PRODQ (JATT, IFILE)
    - FUNCTION GTABL (TAB,X,XLOW, XHIGH, XINCR)
    - SUBROUTINE GDLAY (IFS,ILS,XIN,DEL)

Because of the functions performed by GASP IV, the analyst need only prepare subroutines defining the events and state variables in order to obtain a complete simulation model.

Event Definition

The GASP IV definition of "event" is fundamental to the world-view which supports the modeling of continuous, discrete, or combined systems within the same conceptual framework.

> AN EVENT IS ANY POINT IN TIME BEYOND WHICH THE STATE OF A SYSTEM MAY NOT BE PROJECTED WITH CERTAINTY.

It should be noted that this definition does not relate an event to any change, either discrete or continuous, in the state of a system. Such a relationship often exists, but it is possible to have an event with no associated change in system state. Conversely, it is possible to have a change in system state with no associated event.

In GASP IV, it is useful to describe events in terms of the mechanism by which they are scheduled. Those events which occur at a specified time are referred to as time-events. They are the type of event commonly thought of in conjunction with "next-event" simulation. Events which occur when prescribed conditions defined in terms of the system state are met are called state-events. Unlike time-events, they are not scheduled in the future. They may, however, initiate time-events. Likewise, time-events may initiate state-events. The example presented in this paper illustrates these types of interaction.

The GASP IV Language

The execution of a typical GASP IV program begins with a user provided main program which initiates the simulation. Control is then transferred to GASP, the executive routine, which controls the simulation until completion. A general flow chart of SUBROUTINE GASP is shown in Figure 1.

GASP first calls SUBROUTINE DATIN which initializes all GASP variables either directly or from reading data cards. In addition to initialization, DATIN also provides an echo check of the input data.

Immediately following initialization, GASP prepares to advance simulated time. GASP uses a combined "next-event" and "step-evaluation-step" method of time advance. This combined method is necessary because of the potential existence of state-events whose location on the time axis are not known. GASP first checks to see if there is a time-event to process. If there is, that event is processed by calling SUBROUTINE EVNTS(IX) with the proper event code. If not, GASP checks to see if there are any active state or derivative equations. If there are none, time is advanced from time-event to time-event as each is processed. That is, it proceeds as in GASP II. If there are active state or derivative equations, a different time event mechanism is used. Time is advanced by the maximum allowable step size (user specified) or to the next time-event, whichever is less. (If there are active derivative equations, this involves intermediate steps and accuracy checks.) At that point the system state is examined to see if a state-event has occurred. If a state-event has been passed by more than the specified tolerance, time and state are reset to the beginning of the step and a smaller step size is tried. If no state-events have been passed by more than the specified tolerance all state-events which have occurred within the specified tolerance are processed. If a time-event is scheduled, it is processed. If no event is scheduled, another step is started.

Upon satisfaction of user specified conditions, the run is terminated. GASP then calls SUBROUTINE SUMRY to provide a summary report, calls SUBROUTINE OTPUT to provide user defined output, checks the number of runs remaining, and then either begins a new run or returns to the main program.

Description of Example Problem[1]

As an example of the use of GASP IV, consider the system depicted in Figure 2. A hydrogeneration reaction is conducted in four reactors, each of which may be started, stopped, discharged, or cleaned independently of the others. A compressor with constant molal flow rate provides a supply of hydrogen gas to the reactors. The hydrogen flow is as shown in Figure 2.

---

[1] The authors are indebted to Professor J. M. Woods of the Purdue University School of Chemical Engineering who formulated this example problem.

The operating policy for the facility is to start each of the reactors initially at 30 minute intervals. The concentration of the reactants is then monitored until it reaches 10% of its initial value at which time the reaction is complete and the reactor is turned off. Following completion of a batch, the reactor is discharged, cleaned, recharged and restarted. The time to discharge the reactor is known to be exponentially distributed with a mean of one hour. The time to clean and recharge the reactor is known to be approximately normally distributed with a mean of one hour, a standard deviation of one-half hour, a minimum of zero hours, and a maximum of two hours.

The valve connecting each reactor to the rest of the system is adjusted by controls so as to maintain an effective pressure of 100 psia in each active reactor unless the system pressure has fallen below 100 psia in which case the effective pressure is the actual system pressure.

In order to preclude the pressure from falling too low; if system pressure falls below the critical value (100 psia), the last reactor to have started is immediately shut off. In addition, no reactors are ever started if the system pressure is below the nominal value of 150 psia.

Only two events are associated with the system; a start of reaction event and a stop reaction event. The start of reaction event may be either a time-event, based upon a specified time from completion of one batch to the start of the next batch; or a state-event, based upon system pressure rising above nominal. The stop reaction event will be treated as a state-event, based either upon completion of a batch or pressure falling below critical. These events will be described more fully in the discussion of the coding which follows.

Coding for Example Problem

A liberally annotated listing of the source program for the cited example is given in Figure 3. In large part, the coding is identical to that used in GASP II models, although there is not complete upward compatability. There is however, virtually complete conceptual compatability. Because of this fact, only selected features will be described in this paper. In order to facilitate understanding of the coding, some of the important GASP IV variables are defined below.

ATRIB (I)	Buffer storage for entries being stored in or removed from NSET.
D(I)	The derivative of the Ith state variable.
DTMAX	The maximum step size used to advance time if any state equations requiring integration are active.
DTVG	The difference between TNOW and TLAST.

ID	Maximum number of entries allowed in NSET.
IEVNT	Event code for state-events.
IM	Number of attributes per entry in NSET.
INN(I)	A code establishing the ranking for file I.
IS(I)	A flag indicating the occurrence of a state-event.
JEVNT	Event code for time-events.
KRANK(I)	The attribute number on which file I is ranked.
LSEV	A code indicating whether state-events may cause discrete changes in the system state.
MFA	The relative address of the first space in NSET available for storing a new entry.
MFE(I)	The relative address of the first entry in file I.
MLE(I)	The relative address of the last entry in file I.
NEQD	The number of derivative equations.
NEQDS	The total number of state and derivative equations. (NEQDS=NEQD+NEQS)
NEQS	The number of state equations.
NOQ	The number of separate files in NSET.
NQ(I)	The current number of entries in file I.
NSET(I) QSET(I)	The filing array for storing all entities and their associated pointers.
S(I)	The Ith state variable.
SL(I)	The value of S(I) at TLAST. SL(I) equals S(I) except during periods when GASP is in the process of advancing time.
SEED(I)	The seed for the Ith stream of the random number generator.
TLAST	The latest time at which all of the state variables were completely updated.
TNEXT	The scheduled time of occurrence of the most imminent time-event.
TNOW	Current simulation time.

SUBROUTINE STATE is a required GASP IV subroutine whose purpose is to define the state variables or their derivatives. GASP IV allows substantial flexibility with respect to the definition of state equations. One method of coding subroutine STATE for this example is given in Figure 3. The statements shown below indicate three possible alternative formulations for this problem.

```
1)  S(I)=SL(I)*(1.-DTVG*RK(I)*PEFF*RON(I))

2)  TRL(I)=accumulated running time for
            present batch in reactor I at
            TLAST.

    TR(I)=TRL(I)+RON(I)*DTVG

    XPNTI=TR(I)*RK(I)*PEFF

    S(I)=SO(I)*EXP(XPNTI)

3)  XPNTI=-RON(I)*DTVG*RK(I)*PEFF

    S(I)=SL(I)*EXP(XPNTI)
```

The coding of subroutine STATE and the above alternatives show three general approaches which may be used: 1) Use of the GASP IV provided Runge-Kutta integrator (as in SUBROUTINE STATE); 2) Construction of an Euler integrator (as in

Alternative 1), or; 3) Use of the closed form solution of the problem (as in Alternatives 2 and 3).

SUBROUTINE SCOND performs the dual functions of setting flags to indicate state-event occurrences as well as causing SUBROUTINE GASP to locate any state-events within a prescribed tolerance. The prescribed tolerance may be on the appropriate state variable, on time, or a combination of both.

SUBROUTINE EVNTS(IX) performs the same functions in GASP IV as in GASP II. The only difference is that in addition to being called for each time-event, it is also called for each state-event. For time-events, the argument passed is the event code (JEVNT=ATRIB(2)) of the event to be processed. For state-events, the argument passed is the user specified event code (IEVNT=3 in this example) for state-events. Substantial flexibility exists with respect to making IEVNT a constant or variable and coding the event logic directly into EVNTS or into event subroutines.

SUBROUTINE SEVNT is the state-event subroutine. In this example, it could easily be coded directly in EVNTS, but is separate in order to clarify its function. SEVENT checks those flags set by SCOND and causes the appropriate events to be processed. An alternative method of processing the events, rather than calling the appropriate event routine directly, would be to schedule the event as a time-event to occur at TNOW. That is, to replace CALL START and CALL STOPP by CALL FILEM(1). This approach allows the user to control the sequencing of events which occur at the same instant of time. Thus, simultaneous events may be processed in any user-defined sequence.

SUBROUTINE START describes the performance of the system at the instant in time that the event occurs. If system pressure is below nominal, it causes the entity representing the associated reactor to be filed in the file awaiting conditions enabling the reactor to be started. If system pressure is above nominal it sets the appropriate counters, flags, and attributes and files the entity in the file awaiting conditions causing it to be stopped.

SUBROUTINE STOPP describes the performance of the system at the instant in time that the event occurs. It first sets appropriate flags and counters to indicate the reactor is turned off. Next it checks to see if the STOPP event is caused by batch completion or low pressure. If it is caused by low pressure, attributes are set and the entity is filed awaiting sufficient pressure to start. If it is caused by batch completion, concentration is initialized for the next batch and the start of the next batch is scheduled (the only time-event in this example) for the appropriate time.

SUBROUTINE SSAVE normally does no more than provide a documentation point. It is called at least once at each event time during periods when there are active state equations. If a discrete change in system state may occur at the event time, SSAVE is called both before and after the potential change. Otherwise, SSAVE is called only once at an event time.

Selected Output from Example Problem

The selected output shown in Figures 4, 5 and 6 gives an example of standard GASP output. Several other forms such as error output,event tracing output and state variable tabular output are not shown.

The initial output, Figure 4, consists of an echo check of input data. Definition of those parameters not given previously may be found in Reference 3.

The output shown in Figure 5 is automatically generated by SUBROUTINE SUMRY. Included are tables of parameter values, statistics collected by subroutine COLCT (time each reactor is down after completion of a batch), statistics collected by SUBROUTINE TMST (number of reactors on), a histogram collected by SUBROUTINE HISTO (time each reactor is down after completion of a batch), and a final dump of both the file and state storage areas.

The output shown in Figure 6, is generated by SUBROUTINE GPLOT. In this particular case it provides a plot of each of the state variables as a function of time. The heading lists the user-specified plot symbol and associated identifier as well as the scale for each variable to be plotted. Thus, the plot symbol "P" represents the system pressure on a scale ranging from 0 to 1000 psia. The symbol " · " is used, in this case, to represent critical pressure (100 psia) and nominal pressure (150 psia). Because the plot interval does not equal the communication interval, multiple plot points associated with the same time frequently occur; specifically, where the time step has been refined to locate a state-event or to obtain more accuracy in integration. The dynamic behavior of the system can be seen clearly in Figure 6. Initially, only reactor 1 was on and pressure rose rapidly until reactor 2 was turned on at time 0.5. Reactor 1 was turned off because of batch completion (a state-event) at about 0.7 hours. (More precise accuracy on event times is readily available through either a table giving every event point, a plot with a non-linear time axis which gives every event separately, or a plot with a linear time axis and reduced plot interval.) Beginning with the start of reactor 3 (a time-event) at time 1.0, pressure fell rapidly. There is an obvious discontinuity in the pressure curve at time 1.5, when reactor 4 was started. Pressure first went critical at about 1.8 hours, causing reactor 4 to be stopped (a state-event) since it was the last one started. From 1.8 until 2.8 hours pressure oscillated several times between critical and nominal. (Critical and nominal pressure are shown by the cursors on the plot.) Because pressure falling below critical and pressure rising above nominal are state-events, there is a plot point for each occurrence which greatly aids analysis. It may be noted that the oscillations in pressure caused reactor 4 to be stopped at 1.8, 2.2, 2.5, and 2.8 hours. (Note, the plot point for pressure being critical at time 2.2 coincides with a plot point for reactor 3; thus it is indicated in the duplicates column.)

Applications

Thus far, GASP IV has been used by the authors and by graduate students in a simulation course to code previously published and locally generated models. In each case, the coding has been accomplished without undue difficulty. The previously published models which have been coded in GASP IV include: 1) An Industrial Dynamics formulation of a production-distribution system (Reference 1, pp. 383-386); 2) World Dynamics (Reference 2, pp. 132-134); and, 3) A mechanical impact, Slip Clutch, problem (Reference 4, pp. 74-76). In each case, the GASP IV model replicated the dynamic behavior of the subject model.

References

1. Fahrland, David Arthur. "Combined Discrete Event Continuous Systems Simulation. " Simulation, Volume 14, Number 2, February 1970, pp. 61-72.

2. Forrester, Jay W. Industrial Dynamics. New York: John Wiley and Sons, Inc., 1961.

3. Forrester, Jay W. World Dynamics. Cambridge: Wright-Allen Press, Inc., 1971.

4. Pritsker, A. Alan B. "The Basics of GASP II: A Tutorial," 1971 Winter Simulation Conference, December 8-10, 1971, Waldorf-Astoria, New York, pp. 474-482.

5. Pritsker, A. A. B. and P. J. Kiviat. Simulation with GASP II: A FORTRAN-Based Simulation Language. Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1969.

6. Tramposch, H. and H. A. Jones, Jr. "Impact Problems Efficiently Solved With 1130 CSMP." Simulation, Volume 14, Number 2, February 1970, pp. 73-79.

NGPSS/6000: A NEW IMPLEMENTATION OF GPSS

Karen Ast, Jerry Katzke, Jim Nickerson, Julian Reitman
Norden, Division of United Aircraft Corporation
Norwalk, Connecticut

Lee Rogin
Naval Air Development Center
Warminster, Pennsylvania

Abstract

A version of the GPSS simulation language, compatible with GPSS V, has been implemented for the CDC 6000 series and is operational at the Naval Air Development Center. In addition, the design goals of NGPSS/6000 were thorough assembly debugging, reduced core requirements, decreased running time, unrestricted use of matrices, and language flexibility to allow flexibility for future enhancements. This implementation seeks to aid both the user unfamiliar with the language with more complete diagnostics; while enabling the experienced user to build very large-scale models and associated data banks. Provisions have been made for future language extensions toward real-time operation, on-line debugging, and a faster execution module for debugged models.

INTRODUCTION

The implementation of a GPSS for the CDC 6000 series provided an opportunity to reevaluate the organization and structure of the language. The goal was to preserve the language syntax in its entirety and to simplify the implementation, obtain rapid execution and enable flexibility for future modifications. In addition, as part of this undertaking, it was felt that the resultant language should be internally consistent, for example, matrices can be referenced anywhere within the language.

The first major departure was to implement NGPSS as a two-phase processor consisting of a translator, the assembly phase, and an interpreter, the execution phase. The assembly phase was written primarily in COBOL with COMPASS routines to carry out operations which would have been awkward in the higher level language.

COBOL was chosen for this phase for:

- character manipulation capabilities required to perform syntax checking
- file manipulations used for input, output and cross reference symbol table generation
- sort capabilities used to structure the numeric entity table
- ease of debugging
- to establish a degree of machine independence

The execution phase consists of three distinct segments. The first is an input module which processes the NGPSS control cards and loads the entity area into core, the second is the execution module which performs the actual GPSS simulation and the third is an output module which generates standard NGPSS statistical output. These and any HELP block subroutines are dynamically loaded into core via the SCOPE 3.3 segmentation loader as needed. The first two modules were coded in CDC assembler, COMPASS, to minimize central processor and core requirements. FORTRAN was not used for the following reasons; awkwardness of partial word manipulations, additional core required for system routines, and slower processing of the relative addresses required for entity operations. The output module was written in FORTRAN. Although core and speed are important, they were not as critical as the ability to transform binary values into their character representations.

The two phases of NGPSS/6000 are run as independent operations with communication via a temporary file. The assembly phase is organized in an overlay structure to reduce CPU time. The execution phase is organized into segments because of the reallocatable entity area and the variable number of subroutines. The incompatibility of these loader techniques was resolved by use of a two step processor. The advantage of this type of organization is that assembly and execution can be run independently.

Consistent with the Navy and Norden experience in the building of large scale GPSS models, this implementation of the GPSS language emphasizes the use of matrices to control and operate the GPSS logic. In this manner, a logic system representing the general case can be used with a variety of matrices providing the data to separate the model logic and data libraries. This generalization is emphasized in this implementation by complete accessibility of matrix references anywhere within the GPSS language.

The assembly phase requires 18,700 decimal words of core plus an area for the GPSS symbol table. A typical model can be assembled in default core, 24,576 decimal words. The execution phase requires 7,600 decimal words of core for the system and 3,700 words of core for the system loader and loader tables.

Besides the above considerations, the following features have been added to enhance the language:

1) The restrictions that have been imposed on the usage of matrix and floating point Standard Numerical Attributes (SNA's) have been lifted. They can therefore be used in function arguments and follower cards, GENERATE block arguments, SPLIT block arguments, etc.
2) The number of matrix rows and columns and the transaction number have been added as legal SNA's.
3) A new auxiliary field has been added to the SPLIT block to permit the option of new transactions to no longer be considered as part of an assembly set so that an ASSEMBLY block does not operate on this transaction. This field "GEN" creates a new transaction copy without adding it to the assembly set of the parent transaction.

4) Unary minus operators and a continuation card have been added to arithmetic variables.

ASSEMBLY

Extensive investigation during the assembly design phase suggested the need to eliminate many of the syntax constraints of previous GPSS implementations and provide the user with additional debugging aids and more thorough error checking.

NGPSS/6000 permits a more extensive use of both symbols and matrices. A block or EQUed symbol may be used anywhere a constant is legal; for example, to specify the rows and/or columns on a matrix definition statement. The extensions to the use of symbols in conjunction with use of the SYN statement gives the user greater flexibility than was previously available. For example,

```
ROWS   SYN   10
CLMNS  SYN   20
1      MATRIX MH,ROWS,CLMNS
```

permits the user to vary the number of rows and columns in multiple MATRIX definition statements by simply changing two cards.

. Another extension for the user of symbolics is in the READ mode where the use of symbols is not limited to those defined in that run. Any symbol defined in the SAVE run is automatically defined and can be used in the READ run; in addition, new symbols may be used and will be assigned the next available number for the particular entity type.

. Because all GPSS variable statements are decoded and translated into Reverse Polish notation, there is no limit on the number or depth of parenthesis.

. Additional optional fields were added to the MATRIX definition statement allowing the user to define, EQU, and RESTORE a matrix within one statement.

Numerous debugging aids have been added to the GPSS language. These include an optional cross-reference of all numerically referenced entities in addition to the standard cross-reference of all symbolically referenced entities, a second symbol listing of all symbolically referenced entities in entity number order within entity type, and a summary of the entity allocation. The first scan of function follower cards has been changed so that the search for follower points terminates at the first legal card type; thus, an error in the number of function points does not cause the entire model nor any part of it to be "lost" in the search for the follower points.

PASS2 of the Assembly Phase was completely redesigned, especially that portion of it which scans and decodes block arguments. Block argument decoding is completely table driven and thus the arguments are examined according to the type of argument expected and more complete error checking is possible. The error message and the argument in which the error occurred are printed in addition to the error number; this eliminates any possible confusion concerning in which argument a particular error was found.

Additional Assembly Phase control options have been incorporated in NGPSS/6000. All comments -- both comment cards and comments on any statement -- can be eliminated from the Assembly Listing; this is for use in particular with classified models. Since a free-format scan of the input images requires extra system time, this preliminary scan is performed only when requested. In addition, the free-format input may be listed before the Assembly Listing and a complete source file of the model in fixed format is written so that the user may assemble from the fixed-format file in future runs.

INPUT

In NGPSS/6000 the functions of the Input Phase have been greatly reduced. This was done by moving many of the tasks previously performed by input, particularly in the processing of blocks, to assembly and others to execution. This reduced much of the overlapping of tasks between assembly and input,and input and execution that are present in other implementations of GPSS and lead to a loss of efficiency in running time and core size. For instance, the source code is scanned and decoded, in assembly, and a binary file rather than a coded file is passed to input. Thus, rather than having to scan and decode an intermediate file, the input module, in the case of blocks, has only to store the binary information in the entity area and GPSS common. Similarly, in the case of GPSS control cards such as INITIAL, CLEAR and RESET, the necessary fields have all been scanned and decoded in assembly and only have to be processed in input. Many of the errors are now caught in assembly rather than in the input phase, thereby reducing wasted processing time. A great deal of overlapping of tasks between input and execution has also been eliminated by having all evaluation of SNA's done in execution. This eliminates the need of having all of the SNA evaluation routines present in the INPUT module, thus saving some core. For instance, rather than having the first transaction of a GENERATE block or JOBTAPE created in INPUT, they are done in execution.

Aside from the elimination of all tasks from input that resulted in an overlapping of efforts between modules, a number of other improvements have been made:

1) The READ/SAVE feature executes faster and more efficiently since all pointers to the entity area and GPSS common are stored as relative addresses. Thus, when a saved model is READ the pointers do not have to be adjusted as they would have to be if they were stored as absolute addresses.
2) The INITIAL card has been changed to permit the initialization of Random Number Generators and the character initialization of matrices. Since in NGPSS/6000 the number of random number seeds is reallocatable, the user can selectively initialize whichever ones he wishes to. Character initialization is very useful for printing out reports.
3) At the end of model execution, input prints out the maximum amount of GPSS common and the size of the entity area used during the run. Thus, the user always knows exactly how much common he has available for future additions to his model, or how much common he can eliminate from his model so that it will require less core.
4) A permanent disk-resident matrix data reference library whose data can be accessed randomly was added. Its implementation required no change to the GPSS language structure, merely the addition of 3 control cards; one to create the data library

(KREATE); another to store data in a permanent library (MSTORE) and a third to retrieve it (RESTORE). In addition, the user can retrieve a matrix from a permanent library by use of the E field of a MATRIX definition card. By use of the data base feature the cost of inputting data into a system can be reduced since the data need only be input once and then can be accessed randomly rather than sequentially. Also, the data base can be accessed and modified independent of a model and thus a given model can be run with a variety of data or vice versa.

In NGPSS/6000 the input phase has been streamlined to provide maximum overall efficiency in terms of running time and core size by moving a number of its functions to other modules, adding some new control cards and expanding on the functions of others.

EXECUTION

In the design of the execution phase, the following factors were established as critical considerations:

1) Central processor usage
2) Total memory needed during execution
3) Ease of program modification

One of the major criticisms of existing versions of GPSS has been the total CPU time needed for execution of a large scale model. The objective was to decrease this requirement while staying within memory constraints. Also needed was a system in which enhancements could be made to the language with a minimum of program revision.

In the internal structure of the execution phase, there are two areas which are critical to execution time. The first is the evaluation of block arguments and the second is the structure of the routine which scans the current events chain. Since GPSS is an interpreter each block argument must be evaluated every time the given block is executed. To increase the speed of execution the assigning of default values to missing arguments was given to the assembly phase where each block is processed only once. Also, within execution, space was reserved for the maximum number of arguments for each block. This meant that the arguments could be evaluated as needed and the amount of duplications in argument evaluation could be decreased. The core allocation for matrix arguments was also reassessed. NGPSS/6000 generates a matrix packet for each matrix argument thus nulifying the need for multiple words for each block argument. (See Figure 1 for core allocation of a typical block.)

Because GPSS is a discrete system simulation language, it must be structured so that every active transaction is operated upon after each change in the status of the system. If many transactions are moving at any time, then a great deal of overhead is used in scanning these events. To decrease this overhead the chains use relative addresses and are linked both forward and backward (displacements off the base). The descriptive transaction bits have been moved from the transaction common area to the fixed area to allow for ease in checking to see if the transaction is active. These features along with the incorporation of all chain headers into a common area compensate for the lack of partial word operations on the CDC 6000.

In order to decrease core requirements and use the CDC 6000's 60-bit word, the GPSS entity area has been redesigned. By compressing these areas from 1 to 4 words of core have been saved per entity. Variable statements have also been redefined. Instead of constructing a pseudo object code, NGPSS/6000 uses a Reverse Polish notation. This type of expression removes the need for temporary storage space within each Variable; instead one common area is used by all Variables as an area for storage of temporary calculations.

The final objective was probably the easiest to accomplish. This was done by completely modularizing the execution phase, for example, one routine to evaluate all block arguments, one routine to calculate the address of all entities. By doing this, the addition of a new block means only coding the block and inserting its block mask into the branch table.

OUTPUT

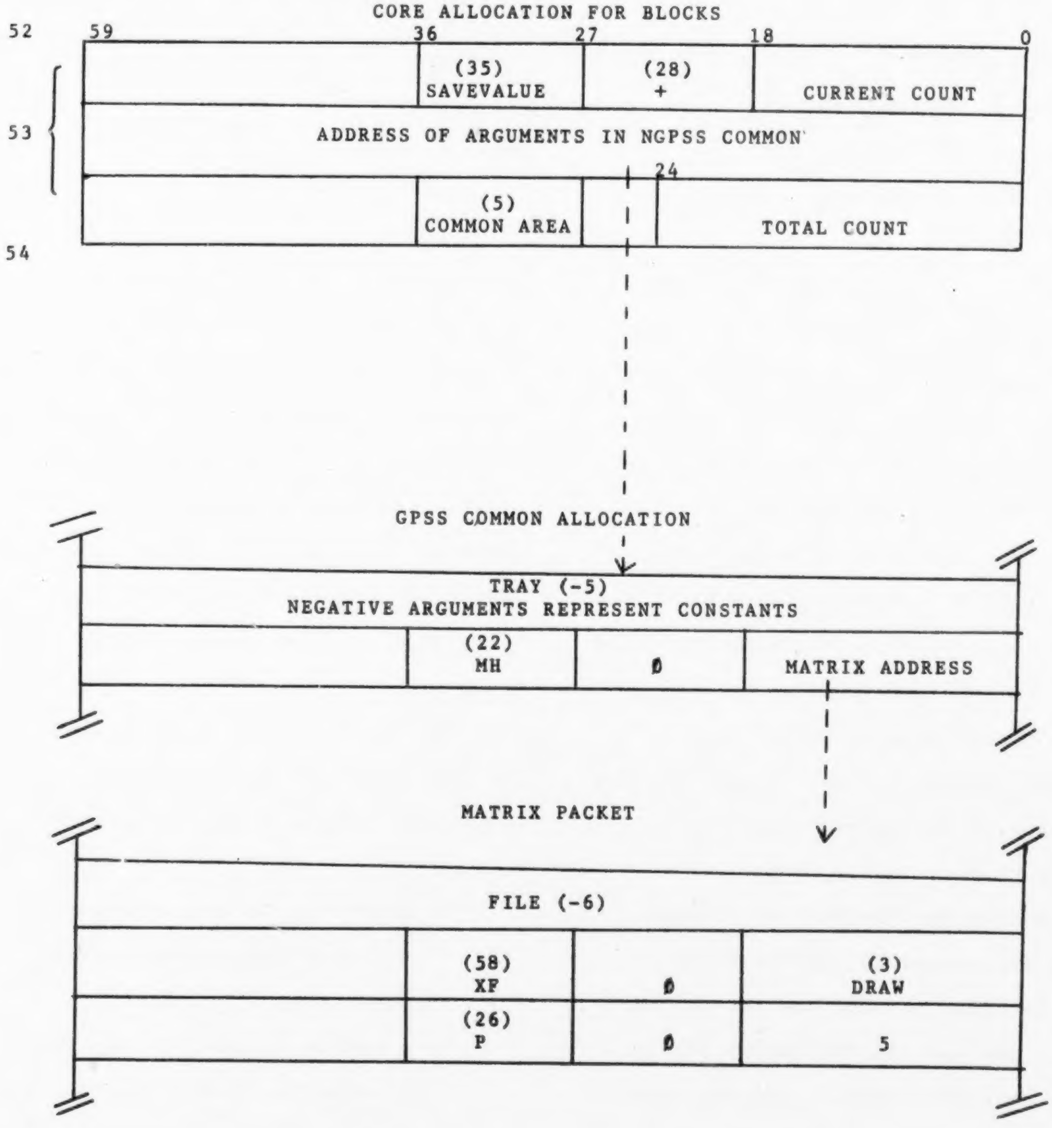
The Output Phase, which prints out the Standard GPSS Statistics, contains a number of minor improvements:

1) Block Symbols are printed out with the block counts where they exist.
2) A random access file is used to retrieve entity symbols thus cutting down on retrieval time.

FIGURE 1

CORE LAYOUT FOR A TYPICAL NGPSS BLOCK

```
53  SAVEVALUE   TRAY+,MH$ FILE (XF$DRAW,P5)
```


CORE ALLOCATION FOR BLOCKS
52
59
36
27
18
0
(35) SAVEVALUE
(28) +
CURRENT COUNT
53
ADDRESS OF ARGUMENTS IN NGPSS COMMON
24
(5) COMMON AREA
TOTAL COUNT
54
GPSS COMMON ALLOCATION
TRAY (-5)
NEGATIVE ARGUMENTS REPRESENT CONSTANTS
(22) MH
0
MATRIX ADDRESS
MATRIX PACKET
FILE (-6)
(58) XF
0
(3) DRAW
(26) P
0
5

3) Only matrix savevalue rows and columns that are non-zero are printed out.

TESTING AND SUMMARY

The testing of NGPSS/6000 has involved a series of models originally coded in GPSS/360 and GPSS V. Since the operation of the pseudo random number generators are machine dependent, a set of models were converted to contain a series of GPSS Variable statements which provided the random number sequence. The running of these models has produced matching statistics and transaction data limited only by the round-off errors in the calculation of utilization factors. As a result of these tests, it is felt that the objective of being able to run GPSS models on a CDC 6000 computer has been achieved.

REFERENCES

1. IBM General Purpose Simulation System V User's Manual, SH-20-0851, November 1970.
2. GPSS/NORDEN Simulation Language, Form 910-1, National CSS, Inc. March 1971.
3. User's Guide to Interactive Simulation (A Superset of GPSS/360), Norden Report #4300 R 0001, August 20, 1970.
4. User's Guide to Conversational GPSS (GPSS/360-NORDEN), Norden Report #4269-R 0003, December 1969.
5. Approaching a Universal GPSS, Jerry Katzke and Julian Reitman, March, 1972.
6. A Computer-Aided Environment for Systems Design, Julian Reitman, October 25, 1971.

CMS/1 - A CORPORATE MODELING SYSTEM

Robert F. Zant

Clemson University

Abstract

Past efforts in the area of corporate modeling have demonstrated the need for "problem-oriented" languages which would facilitate the construction of corporate models. CMS/1 is a discrete-state simulation system designed for use in corporate modeling. It is composed of three languages - a control language, a logic/data specification language, and a report format specification language. Among the features of CMS/1 are 1) the user specification of each variable's name and meaning, 2) the capability of "grouping" variables, 3) the ability to execute a logic module both as a deterministic model and as a stochastic model, and 4) the ability to transfer the values of variables from one model to another.

This paper describes some of the design concepts of the system and presents some examples which illustrate its capabilities.

Introduction

The initial efforts of most firms in building corporate models have followed the philosophy characterized by Dickson, Mauriel, and Anderson as the "fixed structure approach" to model building.[1] Under this approach the model is usually constructed with the aid of a general purpose language such as Fortran or Cobol. The models allow the user little flexibility, forcing him "to employ existing accounts, fixed output reports, and a limited set of options for attaching values to variables in the model."[2]

The limitations of the fixed structure approach have been mitigated by the development of corporate modeling systems such as IBM's PSG,[3] Monsanto's APEX,[4] and Applied Computer Technology Corporation's Foresight.[5] This paper describes a new system, called CMS/1, which was created in an effort to further the development of corporate modeling systems. The logical organization of the system will first be presented and then general design characteristics of the system will be discussed. Third, CMS/1 is briefly compared to another modeling system, Foresight IV. Finally, examples of the use of the system will be given.

Logical Organization

A model is composed of three types of modules. They specify the logical and mathematical relationships for the model (i.e., the logic module), the data to be used in executing the model (i.e., the data module), and the formats of reports which are to be produced (i.e., the report modules).

When the modules are entered into CMS/1, they are translated into a more conveniently executed form and stored on a magnetic disk (see Figure 1). The user may then specify the logic, data, and report modules that are to be retrieved from the disk, merged into a complete model, and executed (see Figure 2).

The execution of a model is actually accomplished in six steps.

1. The first step is to create a matrix of variables composed of the variables referenced by the specified logic module and report modules.

2. Second, the specified data modules are retrieved in turn, and the values are assigned to the indicated variables.

3. The logic module is retrieved and the specified calculations are performed.

4. If the execution is a part of a sensitivity analysis or a Monte Carlo simulation, the second and third steps are repeated.

5. If requested, the values of selected variables are saved.

6. The requested reports are printed. If no report modules are specified, a default report is printed.

Design Characteristics

The basic design characteristics of CMS/1 may be categorized into three groups: those which promote the ease of use, those which promote flexibility of use, and lastly, technical characteristics.

The major contributant to the ease of use of the system is the utilization of "English-like" languages. The use of new specially designed languages means that the modeler need have no previous programming experience. The use of "English-like" languages also make the languages less cryptic and more easily understood. However, the experienced modeler usually tires of using the complete forms of such "English-like" languages; so, short-forms are provided for the expert.

A second contributant to the ease of use of CMS/1 is the provision of default conditions. The more capabilities provided by a modeling system, the more difficult it is to use since the user must select those things he wishes to do from all the alternatives available to him. Relief from this circumstance is provided by the use of defaults. That is, if no selection is explicitly made where an option exists, a predefined alternative is assumed by the modeling system.

CMS/1 also has extensive error-checking routines which check each statement as it is entered. When an error is detected, a concise though clearly worded error message is produced. Where applicable, the message identifies the character, word, or storage file which generated the error.

The modeler is also assisted in the specification of relationships among variables through the capability of "grouping" variables. For example, the statement

```
GROUP LABOR,MATERIALS,OVERHEAD
    UNDER MFG_EXPENSES
```

creates three variables which may be referenced collectively under the name "MFG_EXPENSES." The use of a group name will elicit different responses depending upon the context of its use.

Assuming that the variables SALES, LABOR, MATERIALS, and OVERHEAD have been previously assigned values, the following statement would cause the values of LABOR, MATERIALS, and OVERHEAD to be summed; the sum to be subtracted from the value of SALES; and the result to be associated with the variable OPERATING_INCOME.

```
OPERATING_INCOME EQUALS SALES
    MINUS MFG_EXPENSES
```

Another use of group names is exemplified by the following two statements.

```
GROUP LABOR, MATERIALS, OVERHEAD
    UNDER MFG_EXPENSES, STANDARDS

MFG_EXPENSES EQUAL SALES TIMES
    STANDARDS
```

The second statement above specifies that the values of the labor, materials, and overhead standards are each to be multiplied by the values of sales giving, respectively, the labor, material, and overhead expenses.

The use of CMS/1 is further simplified by the segmentation of models into logic, data, and report modules. The modules are created separately and are combined into one complete model only when executed. A model may contain only one logic module, but any number of data and/or report modules may be used. Thus, a model can be easily altered simply by altering the "mix" of logic, data, and report modules.

This modularization of models also increases the flexibility of the modeling system in several ways. One way is by allowing logic modules to be created independently of the number of time periods over which they are to be executed. The time horizon of a model is limited by the interest of the manager or by the availability of data. When either of these factors change, the logic module may be executed over a longer horizon without altering the module itself. This also means, of course, that different types of logic modules (e.g., long-range planning and capital budgeting models) utilizing different time horizons may be created with CMS/1.

Another implication of the modularization of models is that a logic module may be created independently of its use as a simple deterministic model, its use in sensitivity analysis, and its use in Monte Carlo simulations. A logical relationship may be expressed so that it is invariant over these three situations. It is the data and the procedure used in executing the model which must change.

The segmentation of models is complemented by the facility of referencing all variables by user-supplied names (32 characters per name maximum). The modeler simply references the same variable by the same name in all modules. Then, when the modules are combined for execution, the modeling system links all like names to the same values. This frees the modeler from the burden of maintaining positional equivalency among variables over all modules.

Finally, the flexibility of CMS/1 is greatly enhanced by the facilities for including arithmetic calculations in data modules and for the superceding of selected values calculated in a logic module by values contained in a data module. These two facilities are useful in temporarily altering the logical structure of a model without actually altering the logic module. The alterations are accomplished simply by adding to the "mix" of modules a data module incorporating the desired changes.

There are two technical considerations which have greatly influenced the design of CMS/1. The first is the consideration of the capacity of the system in terms of the number of variables, time periods, etc. which may be accommodated by the system. In some cases no limits are required. There is no direct limit, for example, on the number of data and/or report modules which may be included in a model. In other cases the items must be counted so that the capacity is limited by the maximum value which the counters can obtain. In these instances the capacity of the counters have been set so that the modeler is more likely to be restricted by the physical capacity of his computer than by the capacity of CMS/1. This is exemplified by the capacity of the matrix which is used to store the values of variables. The matrix is dynamically allocated with a maximum capacity of 32,767 variables defined over a maximum of 32,767 time periods. Thus, CMS/1 can accommodate over one billion values.

The second technical consideration deals with the procedure used in executing a model over several time periods. Since CMS/1 is an interpreter rather than a compiler, the most efficient procedure for executing a model would be to execute each operation within a statement over all time periods before proceeding to the next operation. Thus, in executing the statement

```
INTEREST_EXPENSE EQUALS DEBT TIMES
    INTEREST_RATE
```

over five time periods, the multiplication would be carried out five times; and then the five products would be assigned to the variable "INTEREST_EXPENSE."

This procedure, though efficient, yields undesirable results in three cases: 1) when a variable is lagged on itself, 2) when a variable is a function of another lagged variable whose defining equation follows the equation being evaluated, and 3) when a conditional branch occurs. All three of these problems can be avoided by the slower process of interpreting the complete model once for each time period.

When one or more of these three cases arises, the slower procedure is automatically used by CMS/1. In all other cases the more efficient procedure is used by executing, in turn, each operator within an equation over all time periods.

Comparison of CMS/1 and Foresight IV

CMS/1 was originated as an experimental language - one which would offer advanced facilities (e.g., Monte Carlo simulations) for corporate modeling but still be easily used by a novice in computer programming. The resulting design of CMS/1 differs fundamentally from the structure of Foresight. In the Foresight system, a model is basically a single unit containing data, logic, and report formats with the report formats dictating the structure of the model. That is, the data and logic statements are basically algebraic representations of the lines in a report.

In CMS/1 a model is composed of three types of related but separable segments (i.e., data, logic, and reports). The modeler constructs each segment individually. The system then merges the individual segments into a complete model. This segmentation allows the sequence of computations to be independent of the sequence in which the variables appear in a report. Also, one segment can be altered or restructured without necessarily requiring that the other segments be changed.

In addition to the fundamental difference in the approaches of CMS/1 and Foresight to the construction of models, there are also a number of structural differences. CMS/1 has more liberal capacity constraints on items such as the maximum number of variables and periods which may be used, the lengths of statements, or the number of operations in an arithmetic statement. CMS/1 also offers more flexibility in specifying data values and in specifying the form of reports (column widths, etc.). Foresight, on the other hand, has more built-in functions, supports the specification of relationships among columns, and, unlike CMS/1, is available in a time-sharing environment. These and other differences along with some similarities between the two systems are summarized in Table 1.

Example Models

The use of CMS/1 in constructing models is demonstrated in this section through the presentation of models which culminate in the production of a corporate income statement for a hypothetical firm. The sole purpose of the model is to demonstrate the use of CMS/1. They are not intended to reflect the circumstance in any particular firm.

The hypothetical firm is assumed to be composed of two divisions each producing two products. The corporate income statement is therefore developed from the divisions' income statements which in turn depend on the performance of the product lines.

The model for each division is composed of common logic and report modules (see Figure 3) which are combined with different data modules for each division (see Figure 4). The divisional models combine cost and revenue information concerning the product lines (such information is contained in the data modules named "PRODUCT_11", etc.) with information concerning the expenses incurred at the divisional level in order to derive the net contribution of each division. The calculated values are then saved for later use in the corporate level model. (Note that the results from one model may be saved and used as data for another model.) The results of the execution of the two divisional models are depicted in Figures 5 and 6.

The information saved from the divisional models is combined with data on corporate expenses in order to derive a corporate income statement. A complete corporate level model including logic, data, and report modules is depicted in Figure 7. The corporate income statement developed by the model is presented in Figure 8.

Summary

CMS/1 is a corporate modeling system designed to assist in the construction and solution of discrete-state, case-study type models. The emphasis in its design is on the alleviation of the programming burden rather than on the efficient execution of a model.

Among the facilities of CMS/1 are data maintenance services, several special purpose functions such as present value and discounted rate of return computations, and capabilities for performing sensitivity analyses and Monte Carlo simulations. These and other facilities of CMS/1 are currently being extended and improved as a result of experience gained from the utilization of the system by several organizations.

The primary purpose of the development of CMS/1 was to further the development of corporate modeling systems which "will allow the modeler and planner to conceptualize the simulation model in the language it is to be programmed."[6]

Footnotes

1) G.W. Dickson, J. J. Mauriel, and J. C. Anderson, "Computer Assisted Planning Models, A Functional Analysis," Corporate Simulation Models, A. N. Schrieber (ed.), Graduate School of Business Administration, University of Washington, Seattle, Washington, 1970, pp. 43-70.

2) Ibid., p. 53.

3) "Planning Systems Generator User Guide," Program Information Department, International Business Machines Corporation, 1968.

4) Donald L. Buchman, "An Application-Oriented Computer Language for Financial/Economic Simulation," a

paper presented at the XIX International Meeting of the Institute of Management Science, Houston, Texas, April 5, 1972.

5) "Foresight III User Manual," Applied Computer Technology Corporation, Los Angeles, California, 1971.

6. James L. McKenney, "Guidelines For Simulation Model Development," Information Systems Science and Technology, 1967, pp. 169-173.

TABLE 1

Comparison of CMS/1 and Foresight IV

Characteristic	CMS/1	Foresight IV
Batch	Yes	Yes
Time-Sharing	No	Yes
Partition Size	140K +	105K
Computers Used	IBM 360-370 DOS and OS	IBM 360-370 DOS and OS
Sensitivity Analysis	Yes	Yes
Monte Carlo	Yes	No
Arithmetic by Columns	No	Yes
Report Format	Very flexible	Less flexible
Number of Reports	No limit	5
Number of Variables in Model	32,767	1,000
Number of Periods	32,767	21
Statement Length	32,767 characters	160 characters
Complexity of Arithmetic Statement	No limit	Simple with 6 variables and 5 operations maximum
Availability of Functions (present value, etc.)	Several available	More are available
Consolidation of Results from Models	Variables referenced by name	Variables referenced by position in model

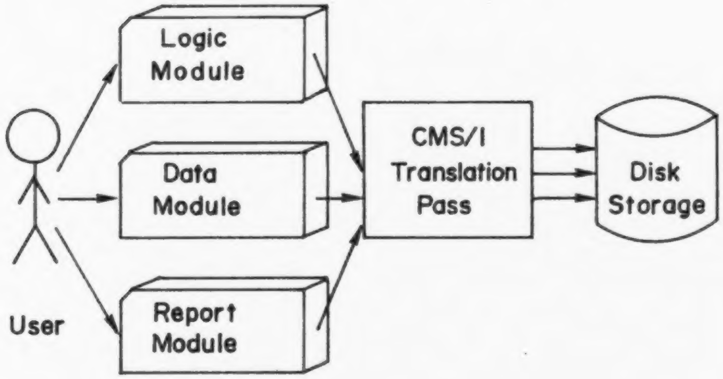

Logic Module
Data Module
Report Module
User
CMS/1 Translation Pass
Disk Storage

Figure 1. Logical organization of CMS/1 - translation pass.

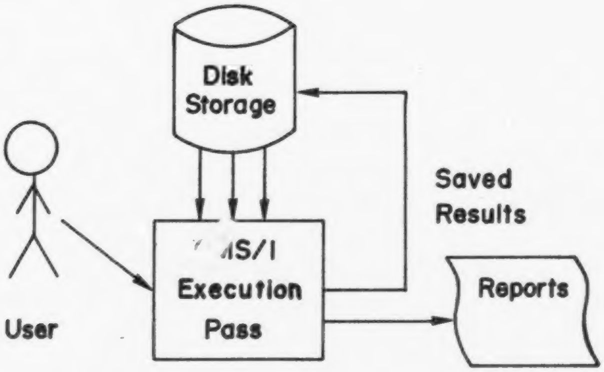

Disk Storage
Saved Results
User
CMS/1 Execution Pass
Reports

Figure 2. Logical organization of CMS/1 - execution pass.

```
*LOGIC  DIVISION
 GROUP  MFG_OVERHEAD, R_AND_D, SELLING_AND_ADVERTISING,
1       ADMINISTRATION  UNDER DIVISION_EXPENSES
 SALES EQUAL SALES1 PLUS SALES2
 MARGINAL_COST = MARGINAL_COST1 + MARGINAL_COST2
 FIXED_PRODUCT_EXPENSES = TOTAL_FIXED_EXPENSES1 PLUS
1                         TOTAL_FIXED_EXPENSES2
 TOTAL_PRODUCT_COSTS = MARGINAL_COST +
1          FIXED_PRODUCT_EXPENSES
 CONTRIBUTION_FROM_PRODUCTS = SALES - TOTAL_PRODUCT_COSTS
 TOTAL_DIVISION_EXPENSES = SUM OF DIVISION_EXPENSES
 CONTRIBUTION_FROM_DIVISION = CONTRIBUTION_FROM_PRODUCTS -
1          TOTAL_DIVISION_EXPENSES
 SAVE  SALES"DIVISION_NO, TOTAL_PRODUCT_COSTS"DIVISION_NO,
1      TOTAL_DIVISION_EXPENSES"DIVISION_NO,
1      CONTRIBUTION_FROM_DIVISION"DIVISION_NO
*REPORT  DIVISION
 TITLE                    DIVISIONAL QUARTERLY PLAN
 MARGIN 0
 LINE LENGTH 60
 COLUMN SIZES  26, 0, (8)
 BEGIN NEW PAGE
 SKIP 2 LINES
 COLUMN HEADINGS  "ACCOUNT", " FIRST", " SECOND",
1                 " THIRD", " FOURTH"
 COLUMN HEADINGS  "--------", " -----", " ------",
1                 " -----", " ------"
 ITEM SALES
 SKIP 1 LINE
 LINE  PRODUCT COSTS:
 ITEM MARGINAL_COST
 ITEM "FIXED COST", FIXED_PRODUCT_EXPENSES
 ITEM CONTRIBUTION_FROM_PRODUCTS
 SKIP 1 LINE
 LINE  DIVISIONAL COSTS:
 ITEM "FIXED MFG. OVERHEAD", DIVISION_EXPENSES:MFG_OVERHEAD
 ITEM "R & D EXPENSES", DIVISION_EXPENSES:R_AND_D
 ITEM DIVISION_EXPENSES:SELLING_AND_ADVERTISING
 ITEM DIVISION_EXPENSES:ADMINISTRATION
 ITEM TOTAL_DIVISION_EXPENSES
 SKIP 1 LINE
 ITEM CONTRIBUTION_FROM_DIVISION
```

Figure 3. Logic and report modules for divisional models.

```
*DATA  PERIODS 1 TO 4
 GROUP  MFG_OVERHEAD, R_AND_D, SELLING_AND_ADVERTISING,
1       ADMINISTRATION  UNDER DIVISION_EXPENSES
 DIVISION_NO IS 1
 DIVISION_EXPENSES:MFG_OVERHEAD = 200000
 DIVISION_EXPENSES:R_AND_D =  50000
 DIVISION_EXPENSES:SELLING_AND_ADVERTISING = 250000
 DIVISION_EXPENSES:ADMINISTRATION = 150000
*EXECUTE  LOGIC DIVISION, DATA PRODUCT_11 PRODUCT_12,
1         SAVE DIVISION_1
1         REPORT DIVISION, HEADING "DIVISION 1"
*DATA  PERIODS 1 TO 4
 GROUP  MFG_OVERHEAD, R_AND_D, SELLING_AND_ADVERTISING,
1       ADMINISTRATION  UNDER DIVISION_EXPENSES
 DIVISION_NO IS 2
 DIVISION_EXPENSES:MFG_OVERHEAD = 100000
 DIVISION_EXPENSES:R_AND_D =  50000
 DIVISION_EXPENSES:SELLING_AND_ADVERTISING = 200000
 DIVISION_EXPENSES:ADMINISTRATION = 150000
*EXECUTE  LOGIC DIVISION, DATA PRODUCT_21 PRODUCT_22,
1         SAVE DIVISION_2
1         REPORT DIVISION, HEADING "DIVISION 2"
```

Figure 4. Data modules and execute statements for divisional models.

```
                          DIVISION 1

                  DIVISIONAL QUARTERLY PLAN

           ACCOUNT               FIRST   SECOND   THIRD   FOURTH
           -------               -----   ------   -----   ------
SALES                           5100000 5100000 4320000 4860000

PRODUCT COSTS:
MARGINAL COST                   3210000 3194000 2704000 3058000
FIXED COST                       800000  800000  800000  800000
CONTRIBUTION FROM PRODUCTS      1090000 1106000  816000 1002000

DIVISIONAL COSTS:
FIXED MFG. OVERHEAD              200000  200000  200000  200000
R & D EXPENSES                    50000   50000   50000   50000
SELLING AND ADVERTISING          250000  250000  250000  250000
ADMINISTRATION                   150000  150000  150000  150000
TOTAL DIVISION EXPENSES          650000  650000  650000  650000

CONTRIBUTION FROM DIVISION       440000  456000  166000  352000
```

Figure 5. Results for first division.

```
                          DIVISION 2

                  DIVISIONAL QUARTERLY PLAN

           ACCOUNT               FIRST   SECOND   THIRD   FOURTH
           -------               -----   ------   -----   ------
SALES                           5550000 5120000 4680000 5060000

PRODUCT COSTS:
MARGINAL COST                   4035000 3632000 3324000 3674000
FIXED COST                       675000  675000  675000  675000
CONTRIBUTION FROM PRODUCTS       840000  813000  681000  711000

DIVISIONAL COSTS:
FIXED MFG. OVERHEAD              100000  100000  100000  100000
R & D EXPENSES                    50000   50000   50000   50000
SELLING AND ADVERTISING          200000  200000  200000  200000
ADMINISTRATION                   150000  150000  150000  150000
TOTAL DIVISION EXPENSES          500000  500000  500000  500000

CONTRIBUTION FROM DIVISION       340000  313000  181000  211000
```

Figure 6. Results for second division.

```
*LOGIC  CORPORATION
 GROUP  ADVERTISING, ADMINISTRATION, OTHER UNDER
1       CORPORATE_EXPENSES
 SALES = SALES1 + SALES2
 PRODUCT_COSTS = TOTAL_PRODUCT_COSTS1 PLUS
1                TOTAL_PRODUCT_COSTS2
 DIVISION_EXPENSES = TOTAL_DIVISION_EXPENSES1 PLUS
1                    TOTAL_DIVISION_EXPENSES2
 CONTRIBUTION_FROM_DIVISIONS = CONTRIBUTION_FROM_DIVISION1
1                         PLUS CONTRIBUTION_FROM_DIVISION2
 TOTAL_CORPORATE_EXPENSES = SUM OF CORPORATE_EXPENSES
 NET_INCOME_BEFORE_TAXES = CONTRIBUTION_FROM_DIVISIONS -
1        TOTAL_CORPORATE_EXPENSES
 IF NET_INCOME_BEFORE_TAXES < 0  THEN JUMP TO NO_TAX
 IF NET_INCOME_BEFORE_TAXES > 25000 THEN TAXES = .48 TIMES
1      NET_INCOME_BEFORE_TAXES - 6500
1                                   ELSE TAXES = .22 TIMES
1      NET_INCOME_BEFORE_TAXES
 JUMP TO AFTER_TAX
 NO_TAX) TAXES = 0
 AFTER_TAX)
 NET_INCOME_AFTER_TAXES = NET_INCOME_BEFORE_TAXES - TAXES
*REPORT  CORPORATION
 TITLE                    CORPORATE QUARTERLY PLAN
 MARGIN 0
 LINE LENGTH 60
 COLUMN SIZES  24, 0, (9)
 BEGIN NEW PAGE
 SKIP 2 LINES
 COLUMN HEADINGS  "ACCOUNT", " FIRST", " SECOND",
1                 " THIRD", "  FOURTH"
 COLUMN HEADINGS  "-------", " -----", " ------",
1                 " -----", "  ------"
 ITEM SALES
 SKIP 1 LINE
 ITEM PRODUCT_COSTS
 ITEM DIVISION_EXPENSES
 LINE  CONTRIBUTION FROM
 ITEM " DIVISIONS", CONTRIBUTION_FROM_DIVISIONS
 SKIP 1 LINE
 LINE  CORPORATE EXPENSES:
 ITEM CORPORATE_EXPENSES:ADVERTISING
 ITEM CORPORATE_EXPENSES:ADMINISTRATION
 ITEM CORPORATE_EXPENSES:OTHER
 ITEM TOTAL_CORPORATE_EXPENSES
 SKIP 1 LINE
 LINE  NET INCOME
 ITEM "  BEFORE TAXES", NET_INCOME_BEFORE_TAXES
 SKIP 1 LINE
 ITEM TAXES
 SKIP 1 LINE
 LINE  NET INCOME
 ITEM "  AFTER TAXES", NET_INCOME_AFTER_TAXES
*DATA  PERIODS 1 TO 4
 GROUP  ADVERTISING, ADMINISTRATION, OTHER UNDER
1       CORPORATE_EXPENSES
 CORPORATE_EXPENSES:ADVERTISING = 100000
 CORPORATE_EXPENSES:ADMINISTRATION = 150000
 CORPORATE_EXPENSES:OTHER =  50000
*EXECUTE  LOGIC CORPORATION, DATA DIVISION_1 DIVISION_2
1         REPORT CORPORATION
```

Figure 7. Complete corporate level model.

```
                         CORPORATE QUARTERLY PLAN

         ACCOUNT              FIRST   SECOND    THIRD   FOURTH
         -------              -----   ------    -----   ------
SALES                      10650000 10220000  9000000  9920000

PRODUCT COSTS               8720000  8301000  7503000  8207000
DIVISION EXPENSES           1150000  1150000  1150000  1150000
CONTRIBUTION FROM
 DIVISIONS                   780000   769000   347000   563000

CORPORATE EXPENSES:
ADVERTISING                  100000   100000   100000   100000
ADMINISTRATION               150000   150000   150000   150000
OTHER                         50000    50000    50000    50000
TOTAL CORPORATE EXPENSES     300000   300000   300000   300000

NET INCOME
  BEFORE TAXES               480000   469000    47000   263000

TAXES                        223900   218620    16060   119740

NET INCOME
  AFTER TAXES                256100   250380    30940   143260
```

Figure 8. Results for corporate level model.

Session 15: Maintenance & Reliability Models
Chairman: Richard E. Barlow, University of California

Three models are presented in which stochastic simulation models are used to construct simulations of systems composed of unreliable components. The releability of the resultant systems is inferred from behavior of these simulation models.

Papers

"A Reliability Model Using Markov Chains for the Utility Evaluation of Computer Systems Onboard Ships"
Carsten Boe Tor Heimly and Tor-Chr. Mathiesen, Det Norske Veritas

"Monte Carlo Simulation of Crosstalk in Communication Cables"
Aridaman K. Jain, Bell Telephone Laboratories, Inc.

"Incorporation of False Alarms in Simulations of Electronic Receivers"
V. P. Sobczynski and C. J. Pearson, SYCOM, Inc.

A RELIABILITY MODEL USING MARKOV CHAINS FOR UTILITY EVALUATION OF COMPUTER SYSTEMS ONBOARD SHIPS

Carsten Bøe, Tor Heimly and Tor-Christian Mathiesen
Det norske Veritas
Oslo, Norway

Abstract

Introduction of computers onboard ships to provide a high degree of automation necessitates calculation of computer system reliability to evaluate the utility of the system. The reliability aspect of the system is simulated by a model using Markov chains. Having defined the system state space and the transition rates, the model provides evaluation of the state probabilities. Evaluation of system utility is based on computer task values and the failure probabilities. Application of the analysis model to an existing system reveals information useful in assigning redundancy, eliminating bottle-necks and allocating spare parts.

INTRODUCTION

The trend towards still higher degrees of automation of machinery plant functions, has increasingly involved the electronic computer as an important active device onboard ships. Primarily, the computer is used to perform monitoring tasks as alarm and safety functions, but also to perform functions as condition monitoring of important components within the machinery plant, and active on-line tasks as bridge control functions, hull monitoring functions and loading/unloading calculations, to mention a few.

Regarding the safety of a ship, one of the most interesting aspects of computerized ship functions is the supervision and automatic control of the machinery plant and especially of the propulsion machinery. In this respect, the requirements of Det norske Veritas as a ship classification society should be mentioned. Already in 1965, Det norske Veritas introduced as the first classification society, rules applying to the instrumentation of machinery plants, intended for periodically unattended operation. These rules are now extended to cover computer installations as well, and in this respect, reliability analysis has proved to be a useful tool.

The introduction of computers onboard ships poses new problems to be considered. The two most important problems are respectively integration of alarm and safety functions in the computer system and the complex environment which is encountered.
The latter problem mostly concerns installation techniques and environmental testing, however, the first problem is of a more philosophical nature regarding systems analysis and design. In this context it is felt that the reliability characteristics of computer hardware alone is not a satisfactory measure of system utility. It is therefore proposed that an approach where the computer system is considered as an integral part of the ship is more realistic when evaluating the utility of a computer system.

The proposed analysis method combines conventional reliability calculations and risk value evaluation into a utility simulation of the computer system, based on the operational characteristics. An important part of the analysis is the establishing of a model of the computer system.

METHOD OF ANALYSIS

The proposed analysis method is intended to be a simple and practical tool in evaluating the utility of shipborne computer systems. The main features are the simplicity of the analysis and the combination of economic and reliability characteristics to provide a better basis for decision making.
Fig. 1 shows some of the elements contained in the analysis.

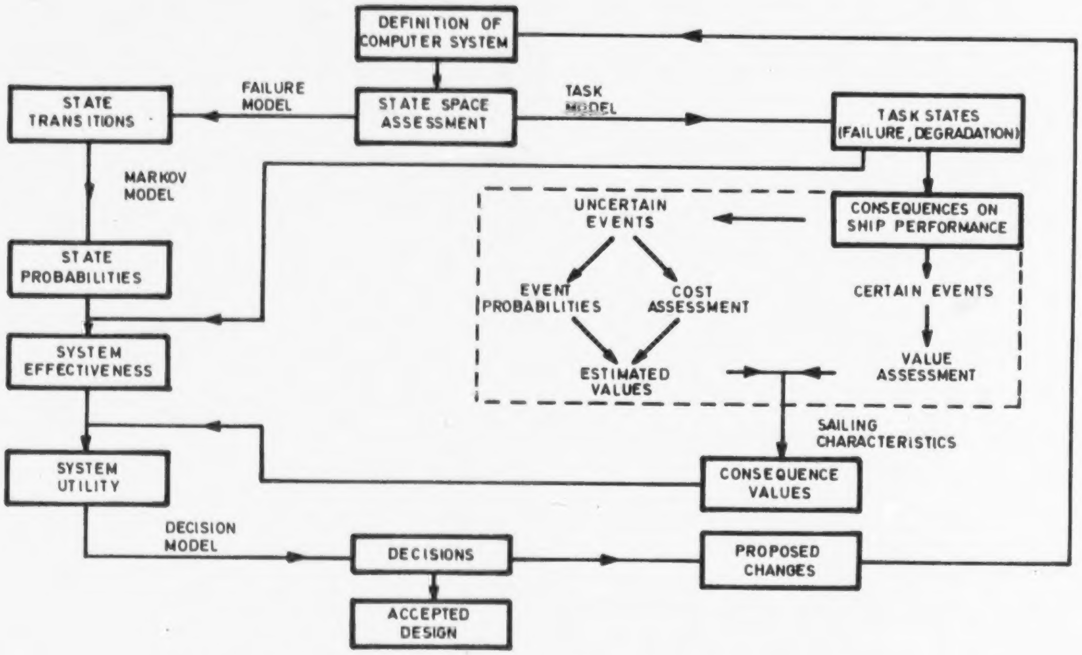

DEFINITION OF COMPUTER SYSTEM
STATE TRANSITIONS
FAILURE MODEL
STATE SPACE ASSESSMENT
TASK MODEL
TASK STATES (FAILURE, DEGRADATION)
MARKOV MODEL
UNCERTAIN EVENTS
CONSEQUENCES ON SHIP PERFORMANCE
STATE PROBABILITIES
EVENT PROBABILITIES
COST ASSESSMENT
CERTAIN EVENTS
SYSTEM EFFECTIVENESS
ESTIMATED VALUES
VALUE ASSESSMENT
SAILING CHARACTERISTICS
SYSTEM UTILITY
CONSEQUENCE VALUES
DECISION MODEL
DECISIONS
PROPOSED CHANGES
ACCEPTED DESIGN

Fig. 1. Procedure for utility analysis of ship computer systems.

The probabilistic approach

Loss of ability to perform the intended tasks constitutes failure of the computer system. Failure of any unit or any combination thereof which can have this effect is considered to be a critical failure. However, failure in some units may only degrade system performance or reduce the instantaneous availability. The computer system thus enters different operational states, depending on the operational states of the units within the system or the transitions between these.

The dynamic behaviour of the system capability may be considered to constitute a stochastic process which actually is defined by the failure behaviour of its units. The operational states of the system are determined from the states of the hardware units and their software implementation. In this context, however, software failures are left alone, and the stochastic process only involves failure of the computer system hardware units. The computer software is looked upon as certain characteristics associated to unique or composite hardware states.

Defining a possible set of events implying system failure, it is possible to define a finite system space. The computer system is assumed to be composed of independent units, and each unit is considered to be in only two states; operating or failed. A system consisting of n independent units may therefore enter 2^n different states, and this state space describes the different failure or operational modes of the system. If the probability of being in a state only depends on the previous state and the transition probabilities between these states are independent of time, the behaviour of the system describes a Markov chain. Given the failure model of the intended unit, the Markov chain will simulate the dynamic behaviour of the system, i.e. the transitions between the possible system states and the probabilities of the different operational modes.

Computer task availabilities

The traditional definition of reliability is impractical to apply to a computer system. This definition says that reliability of some device is defined as the probability the device will perform its function without failure for a specified period of time under stated conditions. Because failure of hardware components does not automatically imply system failure or task failure, the concept of task availability is a better measure of system performance than hardware availability. Some of the system hardware states may leave a specified task completely intact, while others may only degrade the task performance. These states will be termed successful states for that particular task. The states completely destroying task performance will be termed unsuccessful or critical states.

Let $P_j(t)$ be a column vector containing the probabilities of the successful states for task j at time t. Let D_j be another column vector containing numbers indicating the degree to which functional requirements of task j can be accomplished in state i, $i \in \langle \text{successful states} \rangle$ Then

$$A_j(t) = P_j^T(t) \cdot D_j \qquad (1)$$

is a measure of the functional availability of computer task j at time t, given a specified software implementation. $A_j(t)$ may also be termed the system effectiveness for task j at time t. Give a mission time T, the system/mission effectiveness for task j is:

$$E_j = \frac{1}{T} \sum_{i=1}^{m} \left[\int_{t_i}^{t_{i+1}} A_{j,i}(t)\, dt \right] ; \quad (2)$$

where mission duration $T = \sum_{i=1}^{m} (t_{i+1} - t_i)$, m is the number of mission phases and t_i is the time at start of mission phase i.

Defining another column vector $\overline{D}_j = U - D_j$ where U is a unit vector, and substituting $\overline{D}_j$ for D_j in equation (1), gives:

$$\overline{A}_j(t) = P_j^T(t) \cdot \overline{D}_j \; : \qquad (3)$$

where $\overline{A}_j$ is a measure of the unavailability of task j at time t.

Tracing failure consequences

Extending the line of thought in evaluating task availabilities to estimate total computer system performance availability by a nondimensional "percentage" value vector like vector D, does not give the information one needs for making decisions on system utility. In addition, the computer system application and operational environment have to be taken into account.

Given the computer system tasks, their degree of back-up and the operational profile of the ship, a value can be assigned to each task by analysing the consequences resulting from the loss of each task. The consequences concern the whole ship and take the shape of events which may be of a more or less disastrous kind. Some of these events are certain, and some are uncertain - the degree of uncertainty depending mainly on sailing characteristics, loading, geographical position and human interaction. The events can be damage to property, loss of time (off-hire) or inconvenience in the form of dispute, loss of reputation etc..

Tracing the possible sequence of events resulting from loss of a specified computer task, can be accomplished by means of a logical consequence diagram. An estimation of the event sequences, their duration and probabilities can be performed without any knowledge of the failure process of the computer system hardware. The consequence analysis is therefore preferably performed by people with an intimate knowledge of sailing ships and ship operation and with access to damage statistics. Usually several tasks can involve the same events, possibly with different probabilities. The different consequence diagrams thereby become coupled to each other.

Evaluation of system utility

All events resulting from computer task failure are supposed to have a value, either instantaneous or time dependent. These values are independent of how the task failures were initiated and they are estimated from the direct costs associated to an event. The more inaccurate status value of the circumstances connected to the event are also considered.

By collecting all events branching out from a task failure in a consequence diagram and making value estimates from event costs, probabilities and waiting times, a specific value is assigned to every computer task. Going back to the concept of task availability or system effectiveness, the task values are connected to the computer hardware as a measure of the failure consequences from hardware.

Defining the concept of utility as a numerical value of the prospect facing someone in a situation given certain assumptions, the task values can be interpreted as the utility of the different computer system hardware states, (1). Taking into account investment costs, the operating costs and the stochastic failure process of the hardware units, enables the analyst to evaluate the total computer system utility. A utility appraisal of the system can also be done without regarding the investment and operating cost.

SIMULATION MODELS

An important part of the analysis is the simulation of the failure behaviour of the computer hardware system. This produces the system state probabilities which are used as input to the system effectiveness calculations.

The failure rate concept

If f(t) denotes the probability density function of a unit, then f(t) dt is the probability that the unit will fail in the time interval(t, t+dt). The probability that the unit will survive for the period (0, t) is then:

$$R(t) = 1 - \int_0^t f(x)dx = \int_t^\infty f(x)dx \quad ; \qquad (4)$$

which means that:

$$- \frac{dR(t)}{dt} = f(t) \quad ; \qquad (5)$$

The failure rate z(t) of the unit may be defined as the conditional probability that the unit will fail in a time interval (t, t+dt), given that it has survived up to time t:

$$z(t) = \frac{f(t)}{R(t)} = \frac{d}{dt} (\ln R(t)) \quad ; \qquad (6)$$

Assuming the hardware units in the computer system to be subject to chance failures only, the failure rate for each unit is assumed constant: $z(t) = \lambda$, and the expression of reliability in equation (4) becomes:

$$R(t) = e^{-\lambda t} \quad , \text{ since } R(0) = 1 \quad ; \qquad (7)$$

Because the conditional probability $z(t) \cdot dt = \lambda \cdot dt$ depends only on dt and is independent of t, the expected life time of a unit, MTTF, is constant at all times and equal to the reciprocal of the failure rate:

$$MTTF = 1/\lambda \quad ; \qquad (8)$$

This implies that if the independent units composing the computer system have exponentially distributed times to failure, then the time to system hardware failure will also be exponentially distributed. For repairable units, the assumption of constant failure rate λ and repair rate μ, means that operating time between hardware unit failures and the time required for repair (MTTR) of each unit composing the system, are exponentially distributed.

The Markov Model

Having defined a state space for the computer system, a Markov process is one whereby the system occupies a certain state and either undergoes a transition from this state to another, or remains in its present state with time homogeneous transition probabilities which only depend on the previous state.

The Markov chain defined by a discrete state space and continuous time parameter is a stocastic model very suitable in describing the behaviour of complex systems.
Let $p_i(t)$ denote the probability that the system is in a state i at time t. For a state space containing a finite and countable number of states N, obviously

$$\sum_{i=1}^{N} p_i(t) = 1 \quad ; \qquad (9)$$

Let $\mathbf{P}(t)$ be a column vector whose elements are the system state probabilities at time t. $\mathbf{P}(t)$ may be called the state vector. The transition probabilities or rates in the Markov chains will consist of the repair and failure rates of the actual computer system as previously defined.
The requirement of time homogenity is fulfilled by the exponential density functions for time to failure and time to repair. Use of the Chapman Kolmogorov differential equation gives

$$\frac{d}{dt}\,\mathbf{P}(t) = (M) \cdot \mathbf{P}(t) \quad : \qquad (10)$$

where (M) is the NxN matrix of the transition rates.

Knowing the initial conditions given by the state vector $\mathbf{P}(0)$, the set of simultaneous differential equations can be solved, and the probability vector for the system states is obtained as a function of time.
The transition rate matrix(M) is the basic element in the Markov model, and it characterizes both the system being analysed as well as the analysis.

If the computer system is repairable in all states containing failed hardware units, i.e. all states communicate, then the transition matrix and the states are called ergodic or positive recurrent. States which are not ergodic, are called transient.

In a completely ergodic process, the limits:

$$\lim_{t \to \infty} P_i(t) = P_i \qquad (11)$$

exist for all states i in the state space. As

$t \to \infty$, equation (10) becomes:

$$(M) \cdot \mathbf{P} = 0 \qquad (12)$$

Together with equation (9) this equation implies that the limiting state probabilities can be determined by solving a set of linear algebraic equations.

A useful tool in Markov analysis is to prepare a diagrammatic representation of the transition rate matrix(M). The graph is called a reliability transition diagram, and it is composed of nodes representing system states and branches representing the possible transitions between the states. Labelling the branches with transition rates makes it quite simple to evaluate the elements in (M). Examples of transition diagrams are given in fig. 4 and fig. 5.

Computer programs

In order to cope with the problems of solving the equation systems of equation (10) and (12 and 9) in a fast and economical manner, two computer programs have been developed, REAVAN and STAVAN (2).

The program REAVAN solves the set of differential equations given by equation (10), utilizing the Kutta-Merson algorithm. The result is the probability state vector P(t) as a function of time, for a finite time period with specified time intervals.

The program STAVAN solves the set of linear algebraic equations given by equations (9) and (12). The solution technique is based on an optimal Jordan elimination process, and the result is the steady-state probability vector P and the waiting times between different specified subsets of states.

Both programs are written in the ALGOL programming language for UNIVAC 1107 and 1108 computers with EXEC 8. Some of the subroutines involving manipulation of matrices are, however, written in FORTRAN IV. The programs have proved to be extremely helpful in evaluating system state probabilities. Computing time being only a few seconds, the programs are economical to run and give a lot of information in short time.

APPLICATION OF THE ANALYSIS METHOD TO A DESIGN STUDY

Given an actual ship and the tasks to be performed by the computer system, an example will be given, showing how use of the described method can be used to increase the utility of a system at the design stage. This is done by assigning redundancy, eliminate bottle-necks and allocate spare parts with respect to the ship's function and environmental conditions.

The ship system and cost values

The ship system to be considered is a machinery plant, with special emphasis placed on the propulsion machinery. Supervision and control of the machinery (referred to as the E0 tasks) and condition monitoring (referred to as CM) are the main tasks to be performed by the computer system.

The analysis method allows partition of the analysis into two groups, or submodels of the overall system. One consists of the computer system including the tasks to be performed. The other is the ship system which defines the computer tasks. Description of the ship system and the assigning of cost values to the different tasks to be performed by the computer system, will not be shown here. The value estimation can best be done by personnel with experience in and knowledge of sailing ships and ship machinery plants, since the value estimation is independent of the computer configuration.

The cost estimation must take account of sailing schedules, harbourage, type of ship etc.. Stop of main propulsion involves greater risk, i.e. expected cost, to the ship when manoeuvring in restricted water than when sailing in open sea.

For the estimation of the different cost values, a typical voyage of 24 days in open sea, 4 days in restricted waters and two days in harbour is taken into consideration, (3).

The time dependent and immediate values for loss of computer tasks are shown in table 1. These values are valid for all four system alternatives outlined in the following.

The basic computer system

The starting point in this design study is the basic computer system A, shown in fig. 2. It consists of a computer (COM) (CPU, memory, interface for typewriter, punch, tapereader, computer operator panel etc.) and a typewriter (TW). Further there is a control console (CC) connected to the computer through the process input/output system (PIO). A display (CRT) is also connected through the PIO. A tapereader (TR) is used for loading programs into the computer, and the whole computer system is fed by power from the main switchboard (MSB). No tapepunch is shown since it is not necessary for the overall system function.

The failure and repair data used in this analysis are estimated after communication with designers of related systems. The main input data to the computer programs REAVAN and STAVAN are the mean time between failure (MTBF) and the mean time to repair (MTTR) for each component in the system. In table 2, the columns 3 and 4 show data valid for the basic system A. (The table also includes data used for the systems B, C and D.)

The basic system is supposed to consist of seven independent units (see fig. 2). Each unit is considered to be in only one of two states, operating or failed. The system may therefore enter $2^7 = 128$ different states. Since the MTBF is much larger than the MTTR for all units in the system, every combination of unit failures yielding consequences less severe than the consequences of each subset within the combination are neglected. Figure 4 shows the reliability transition diagram for the basic system A. Only 9 states are considered to be of interest. The states 2, 3, 4 and 5 will cause loss of all computer tasks.

Some results obtained from the computer program REAVAN are shown in fig. 6, 7 and 8 for all four system configurations. In fig. 6a, the dynamic behaviour of the probabilities of system success are plotted. The steady state availabilities for repairable systems are reached in approximately 8 to 12 hours after starting the systems. Fig. 6b shows the corresponding probabilities of computer failure which is a critical failure mode. In measuring the utility value of the system, a.o. equation (3) is used to compute the task value function for the system. The simple decision table, table 3, shows the connection between hardware failures and total or partial loss of computer tasks. Decision tables are used to prepare information for input to computer programs calculating utility values.

The calculated utility values for loss of system performance are shown in fig. 7. In fig. 8, the task availabilities are shown as calculated by the computer program STAVAN.

Analysis method

The three system configurations B, C and D are modifications of basic computer system A. The objective is to improve the availability of the computer tasks, thus decreasing the overall risk utility. Experience has shown that the power supply from the main switchboard is critical. Use of this power supply causes the MTTF for several units, especially the computer and tape-reader, to decrease to a value much below the corresponding value for land-based computer systems. Feeding power continuously through a battery bank to the computer system, improves the MTTF for several units, see 5th and 6th column in table 2. Additionally, the battery power supply guarantees the system continuous power for at least 30 min. if a main switchboard break-down occurs. The addition of a battery supply to system A gives system B.

Table 4 shows that the E0 and CM tasks depend heavily on precise function of the typewriter. The table also shows that no reconfiguration of the program system can be performed without proper function of the tape-reader. Usually, typewriters are equipped with slow tape-readers. Modifying the software and hardware system in such a way that the tape-reader on the typewriter can be used as back-up, and adding an extra typewriter for redundancy, we call the new configuration system C.

Again, calculations on the modified computer system show an improvement of the utility function in spite of a small decrease in the steady state and dynamic availability of the computer hardware. According to table 5, the only "bottle-necks" remaining are the computer itself and the process input/output system. These are the only units which by a single failure can cause total system break-down.

The failure consequences presented in table 1, show that the E0-tasks are much more important than the CM-tasks. Using two computers, one for the E0-tasks and the other for the CM-tasks, are giving back-up for the E0-computer tasks at the expense of the CM-tasks. This also results in a higher MTTF for each computer in this new system compared to the computers in system A, B and C, owing to reduction of the memory capacity of each computer. Using this modification, a data channel (ACM) is needed for communication between the two computers.

In the process I/O system, the multiplexers and converters are some of the most unreliable parts. The I/O system is divided into three parts. Two identical parts, containing the most unreliable part of the I/O system, serve each of the two computers. The part serving the less important computer will serve as a stand-by unit for the most important computer (E0-tasks). The third part of the I/O system is quite reliable, so this remaining bottle-neck is acceptable from a reliability point of view.

A block diagram for this system, containing 13 units, is shown in fig. 3. Theoretically, the system can enter $2^{13} = 8192$ different states, but without loss of any significant information, the method applied allows for a reduction to only 28 states. The reliability transition diagram for system D is shown in fig. 5.

The improvement in risk utility from system A to system D is shown to be a factor of 2.5 (fig. 7), with a corresponding increase in task availability.

Hitherto, we have assumed that all system failures have been repairable with a MTTR given in table 2. This will not always be true, especially for computer systems onboard ships due to lack of specialists, tools, spare parts etc.

In order to demonstrate a way of allocating spare parts, calculations have been performed assuming that the CRT display, battery power supply and two identical parts of the I/O system are not repairable (absorbing states). The results are plotted into fig. 6a.

CONCLUSION

In the preceding sections, an analysis method intended for evaluation of computer systems on board, has been presented. The procedure may seem somewhat complex at first, but it has been found to be a simple and efficient way of obtaining information on computer system structures. It is felt that reliability data alone are not satisfactory as a basis for selecting between alternative system configurations. The concepts of utility and task values, however, prove to provide information relevant to systems evaluation and design.

Application of Markov models has been found to constitute a very convenient analysis tool in systems design, because:

- The concept is easy to understand.
- The model is easy to use.
- The state space is easy to change.
- The system structure is easy to change.
- Alternative systems are easily compared.
- Sensitivity analysis is easy to perform.
- Computer analysis takes only a few seconds.

Even if the Markov chains in some cases may not be the correct stochastic description of the system, it still gives information enabling comparative analysis of systems.

REFERENCES:

(1) H. Chernoff and L. E. Moses: "Elementary Decision Theory". John Wiley & Sons Inc., New York 1959.

(2) Tor-Christian Mathiesen: "Reliability Engineering and Ship Machinery Plant Design". Lic. Techn. Thesis.

(3) T. Heimly, G. Dahll, C. Bøe: "Reliability and Availability of Computer Systems onboard Ships" (In Norwegian). Report from Det norske Veritas, Machinery department 1972.

	E0	CM
Immediate value	$2.7 \cdot 10^4$	0
Time dependent value	$97 \cdot 10^4$	$0.21 \cdot 10^4$

Table 1. Value for computer tasks.

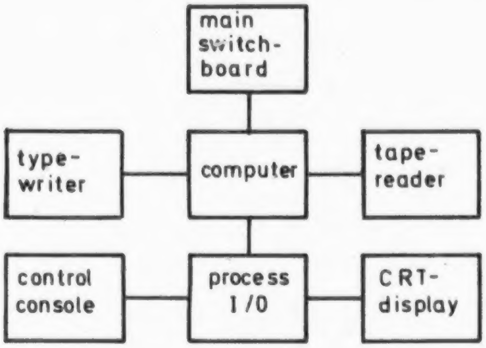

main switch-board
type-writer
computer
tape-reader
control console
process I/O
CRT-display

Fig 2. Basic computer system A

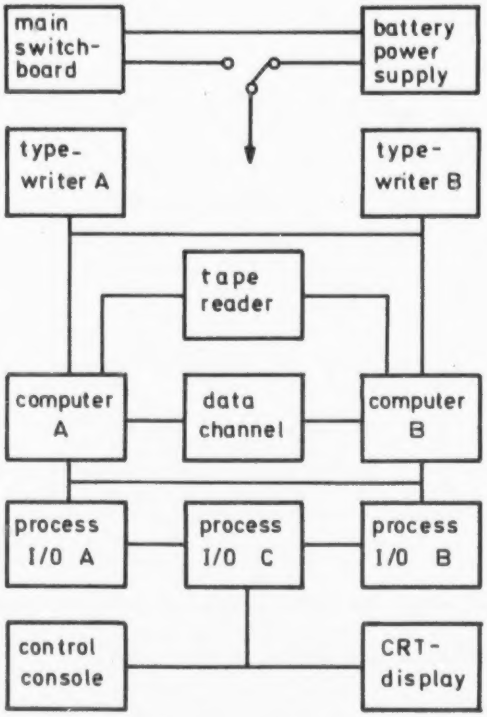

main switch-board
battery power supply
type-writer A
type-writer B
tape reader
computer A
data channel
computer B
process I/O A
process I/O C
process I/O B
control console
CRT-display

Fig. 3. Computer system D

	Component		using Main switchboard		using Battery power supply	
			MTTF (hours)	MTTR (hours)	MTTF (hours)	MTTR (hours)
	Computer (system A, B and C)	COM	1250	4	2000	4
	Typewriter	TW	1000	2	1000	2
	Tapereader	TR	1/2*)	5	1/10*)	5
	Control console	CC	25000	10	30000	10
	Process I/O	PIO	8000	6	10000	6
	CRT-display	CRT	2000	8	3000	8
	Main switchboard	MSB	1000	1	1000	1
	Battery power supply	BPS	1000	3	1000	3
	Computer using tapereader on typewriter			8		10
System D	Data channel	ACM	10000	5	20000	5
	Computer A	COMA	2500	4	5000	4
	Computer B	COMB	2500	4	5000	4
	Process I/O, part A	PIOA	8000	6	10000	6
	Process I/O, part B	PIOB	8000	6	10000	6
	Process I/O, part C	PIOC	40000	6	40000	6

*) failure every 2nd or 10th time when used.

Table 2 Failure and repair data

COM	y						y				ELSE
PIO		y									
MSB			y							y	
TW				y				y			
CC					y						
CRT						y		y			
TR							y				
BPS									y	y	
EO	1	1	0	.2	.4	0	1	.2	.1	1	E
CM	1	1	0	.6	.4	.5	1	.8	.1	1	E

Table 4. Decision table, system B
(E = error, considering not allowed
states)

COM	y											y		ELSE
PIO		y												
MSB			y										y	
TWA				y							y			
TWB					y						y			
TWA or TWB						y								
CC							y							
CRT								y						
TR									y			y		
BPS										y			y	
EO	1	1	0	0	0	0	.4	0	0	.1	.2	.1	1	E
CM	1	1	0	.4	.4	.4	4	.5	0	.1	.6	.1	1	E

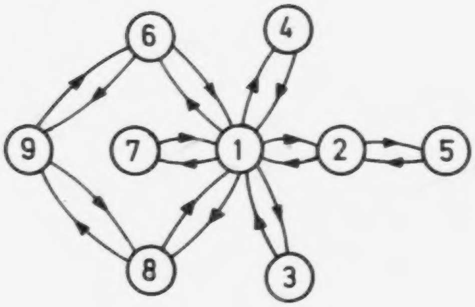
Table 5. Decision table, system C

COMA	y															y			y		ELSE
COMB		y														y					
PIOA			y														y				
PIOB				y													y				
PIOC					y																
PIOA or PIOB						y															
MSB							y													y	
TWA								y										y			
TWB									y									y			
TWA or TWB										y											
CC											y										
CRT												y									
TR													y						y		
ACM														y							
BPS															y					y	
EO	0	0	0	0	1	0	0	0	0	0	.4	0	0	0	.1	1	1	.2	.1	1	E
CM	1	1	1	1	1	1	0	.4	.4	.4	.4	.5	0	1	.1	1	1	.6	.1	1	E

Table 6. Decision table, system D

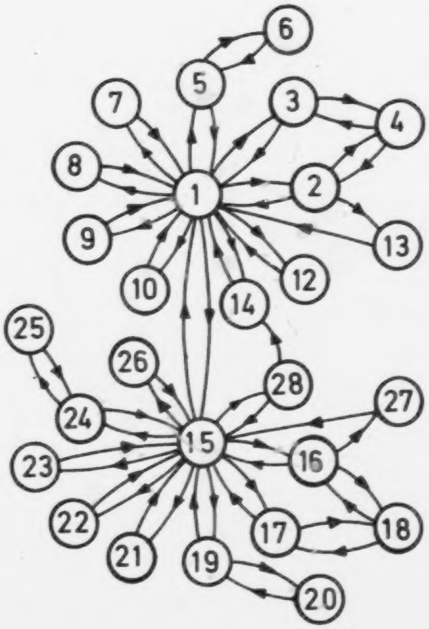
COM	y							y		ELSE
PIO		y								
MSB			y							
TW				y					y	
CC					y					
CRT						y			y	
TR							y	y		
EO	1	1	1	.2	4	0	0	1	.2	E
CM	1	1	1	.6	.4	.5	0	1	.8	E

Table 3. Decision table, system A



1. System o.k.
2. Computer fault
3. Process I/O fault
4. Main switchboard fault
5. Computer and tapereader fault
6. Typewriter fault
7. Control console fault
8. CRT-display fault
9. CRT and typewriter fault

Fig. 4. Reliability transition diagram for system A.



1. System o.k.
2. Computer A fault
3. Computer B fault
4. Computer A and B fault
5. Typewriter A or B fault
6. Typewriter A and B fault
7. Control console fault
8. CRT-display fault
9. Process I/O C fault
10. Process I/O A or B fault
11. Process I/O A and B fault
12. Data channel fault
13. Computer A and tapereader fault
14. Main switchboard fault
15. Battery power supply fault

Battery power supply fault
16. Computer A fault
17. Computer B fault
18. Computer A and B fault
19. Typewriter A or B fault
20. Typewriter A and B fault
21. Control console fault
22. CRT-display fault
23. Process I/O C fault
24. Process I/O A or B fault
25. Process I/O A and B fault
26. Data channel fault
27. Computer A and tapereader fault
28. Main switchboard and battery power supply fault

Fig. 5. Reliability transition diagram for system D.

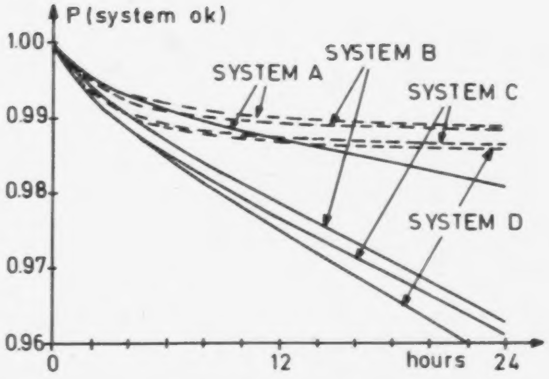


Fig. 6a Dynamic behavior of systems as calculated by REAVAN

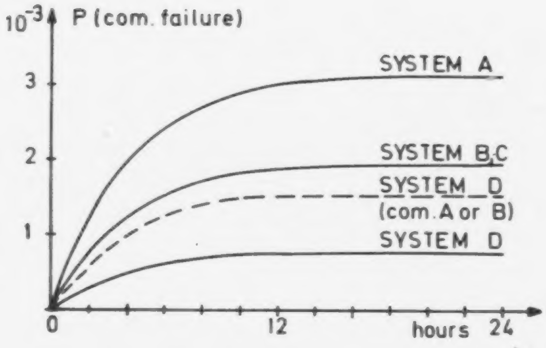


Fig. 6b Probability of computer failure as calculated by REAVAN

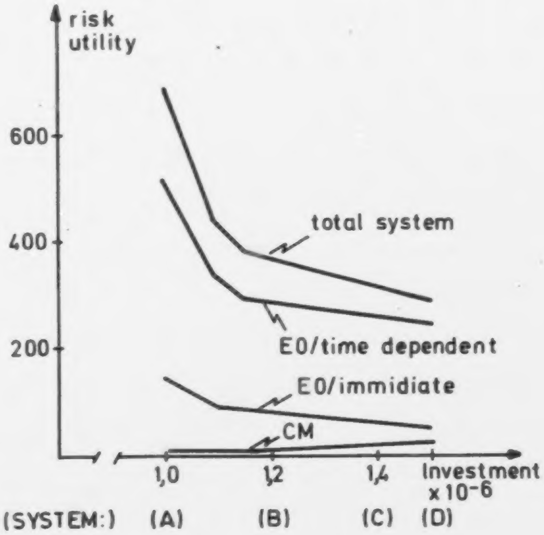


Fig. 7. The utility function

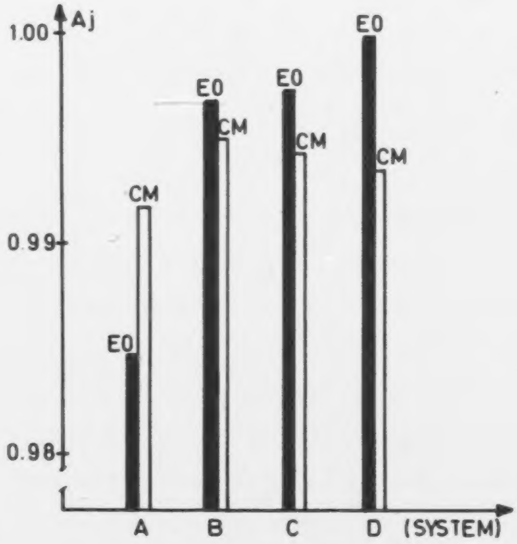


Fig. 8 Task availabilities as calculated by STAVAN

"MONTE CARLO SIMULATION OF CROSSTALK IN COMMUNICATION CABLES"

A. K. Jain

Bell Telephone Laboratories, Incorporated
Holmdel, New Jersey 07733

Abstract

One way of developing a communication cable of new design is to make many experimental cables and to study their path-to-path crosstalk properties. This method is both time-consuming and expensive. In this paper, use is made of evaluating "synthetic" cables. First, we fit statistical models, which are functions of design parameters, to measurements of crosstalk in one or more experimental cable(s). Then we use these models and estimates of associated components of statistical variability in Monte Carlo simulation of crosstalk thereby enabling the exploration of cable designs different from those of the experimental cables.

1. Introduction

Far-end crosstalk is one of the major sources of interference in communication paths. Cable designers would like to develop new communication cables whose path-to-path crosstalk* is at a desired level. One way of developing a communication cable of a new design is to make many experimental cables and to study their path-to-path crosstalk properties. This method is both time-consuming and expensive. Here we describe another approach which develops and evaluates "synthetic" cables.

First we analyze path-to-path crosstalk measurements for experimental cable(s) to estimate components of statistical variability and to fit statistical models, which are functions of design parameters, to these measurements. Then these models are used in Monte Carlo simulation of path-to-path crosstalk for cables with designs different from that (those) for the experimental cable(s). The individual crosstalk into each path (pair) from the remaining paths are "added" to get overall indices of performance called power sums. The power sum distributions of "synthetic" cables are used to evaluate their designs. This evaluation leads to further exploration of new cable designs.

---

* From now on, far-end crosstalk is abbreviated as crosstalk.

2. Data Collection

Statistical analysis of crosstalk measurements from one or more experimental cables is the backbone of the approach described here. An experimental low capacitance cable was designed and manufactured during the course of recent development of a new transmission system. We will use this cable to illustrate the approach. A cross-section of this cable is shown in Figure 1. This cable has 102 pairs and it is made up of six layers. In each layer there is a certain number of distinct twist lengths which are repeated a certain number of times (Table 1). The six layers are stranded (i.e., the pairs in each layer go around the center) alternately in the right and left directions.

The crosstalk measurement equipment was limited physically to measure not more than 50 pairs in a setup and it was feasible to measure only two 50-pair sets in total. The two 50-pair sets chosen for measurement (yielding $2450=2\binom{50}{2}$ pair-to-pair crosstalk measurements) are indicated in Figure 2. These two sets were selected to provide sufficient data for estimating the performance of the current cable design and for exploring other twist length selections in future cable designs.

3. Model Development and Simulation of Crosstalk

3.1 Model Development

Table 2 gives a breakdown of the 2450 measurements into three groups: (i) within layers (i.e., when both the disturbed pair i and the disturbing pair j belong to the same layer), (ii) between layers when $D_{ij} \leq 3$, where

TABLE 1

Twist Length Assignment (inches)

Layer No.	Total No. of Pairs	No. of Distinct Twists	Distinct Twist No.									Str. Lay* (ins.)
			1	2	3	4	5	6	7	8	9	
1	3	3	0.9	1.4	1.2							24R
2	9	9	1.8	2.7	6.2	2.0	3.8	1.7	2.9	5.5	2.2	24L
3	15	3	1.1	1.2	1.5							24R
4	20	5	7.1	3.0	4.6	2.5	3.6					24L
5	25	5	0.9	1.5	1.2	1.8	1.4					24R
6	30	6	6.2	2.9	2.0	4.4	3.4	2.4				24L

---

* An effect of stranding lay is to possibly change relative twist lengths of two pairs. This adjustment is made in defining "effective twist lengths" which are used in the analysis to follow.

TABLE 2

Breakdown of the 2450 Measurements

Group	No. of Pair Combinations*			
	Distinct	With Information on Manuf. Var.	With Repeat Meas.	Total
Within layers	267	104	49	420
Between layers I ($D_{ij} \leq 3$)	301	1294	87	1682
Between layers II ($D_{ij} > 3$)	87	261	0	348
Total	655	1659	136	2450

D_{ij} is the minimum distance† between the two pairs involved, and (iii) between layers when $D_{ij} > 3$. The 17 pairs common in the two 50-pair sets provide repeat measurements which in turn yield information on the measurement variability. Similarly, identification of distinct pair combinations (see Figure 1) and an analysis of different realizations (if more than one) of each distinct pair combination provides an estimate of the variability due to the manufacturing process. The following model expresses crosstalk measurements (in dB) as sums of several components:

$$C_{\underset{\text{Group \#}}{q}\,\underset{\text{Pair Comb. \#}}{\ell}\,\underset{\text{Replicate \#}}{m}\,\underset{\text{Meas. \#}}{n}} = \underset{\text{Overall Mean}}{\mu} + \underset{\text{Group Effect}}{Gq} + \underset{\text{Pair Comb. Effect}}{P_{(q)\ell}} + \underset{\text{Manuf. Var.}}{\delta_{(q\ell)m}} + \underset{\text{Meas. Variation}}{\epsilon_{(q\ell m)n}}$$

where $\delta_{(q\ell)m} \sim N\left(0,\sigma^2_{(q)\delta}\right)$ and $\epsilon_{(q\ell m)n} \sim N\left(0,\sigma^2_{(q)}\right)$. Table 3 shows the mean squares due to (i) distinct pair comb., (ii) manufacturing variability, and (iii) measurement variability in each of the three groups. It is clear that manufacturing variability is quite large.

Next we fitted regression functions which attempt to describe average crosstalk for distinct pair combinations as functions of the following design variables:

D_{ij} = minimum distance between pairs i and j where the unit of distance is the diameter of a pair,

T_i = twist length of pair i (in inches),

T_j = twist length of pair j, and

$$L_i = \begin{cases} 1 & \text{if pair i is in layer 6} \\ 0 & \text{otherwise} \end{cases}$$

A regression function for group q may be written as

$$C_{(q)ij} = f_q(D_{ij},T_i,T_j,L_i)$$

---

* The numbers in the middle two columns are more properly labeled as degrees of freedom.

† The unit of distance is the diameter of a pair.

TABLE 3

A Breakdown of Variation by Group

Group	Mean Square*		
	Distinct Pair Comb.	Manufacturing Variability	Measurement Variability
Within layers	122.23 (266)	33.91 (104)	5.39 (49)
Between layers I ($D_{ij} \leq 3$)	225.91 (300)	42.14 (1294)	6.71 (87)
Between layers II ($D_{ij} > 3$)	129.02 (86)	23.24 (261)	-

where $C_{(q)ij}$ = crosstalk from pair j to pair i in group q.

The functional form of f was explored by plotting $C_{(q)ij}$ against the design variables and some simple functions of these variables. Let us illustrate the fitting procedure by considering the within-layer group. For this group the only variable which indicated a significant relationship with C_{ij} (q, which is equal to 1 for within-layer group, is omitted for convenience) is D_{ij}. In other words, if we consider only one variable at a time (and consequently ignore the other variables), we are not able to detect any relationship of C_{ij} with the other design variables. After fitting $C_{ij} = 57.04 + 18.33 \log(D_{ij})$, and plotting C'_{ij} $\left(= C_{ij} - 57.04 - 18.33 \log(D_{ij})\right)$ against the twist length variables, we noticed a negative relationship with $T_i + T_j$ which led to the new regression:

$$C_{ij} = 61.62 + 18.69 \log(D_{ij}) - 6.64 \log(T_i+T_j).$$

Again we computed the new residuals

$$C''_{ij} = C_{ij} - 61.62 - 18.69 \log(D_{ij}) + 6.64 \log(T_i+T_j) ,$$

and made a plot of C''_{ij} against other variables which suggested the addition of $\log(|T_i-T_j|)^{\dagger}$ and L_i. No additional terms, which make further improvement in the explanation of C_{ij}, could be found. Thus, the best fitted statistical model for within-layer crosstalk is

$$C_{ij} = 64.08 - 3.53L_i + \log_{10}\left\{\frac{D_{ij}^{18.94}\ (|T_i-T_j|)^{2.45}}{(T_i+T_j)^{8.83}}\right\} + e_{ij} \tag{1}$$

where

$$e_{ij} = \delta_{ij} + \varepsilon_{ij} + \text{lack of fit, and}$$

$$e_{ij} \sim N(0,49.05).$$

---

[*] The numbers in parentheses are degrees of freedom.

[†] If $|T_i-T_j| < 0.05$, then we substitute $|T_i-T_j| = 0.05$.

Lack of fit represents the discrepancy between the fitted model and the true relationship (if one exists) between C_{ij} and the design variables. The above fitted model does not fit the average crosstalk for distinct pair combinations perfectly. However, the lack of fit is small and the above fitted model is quite useful in describing crosstalk as a function of design variables.

Similarly, the best fitted models for the between layers groups were the following:

Between layers I ($D_{ij} \leq 3$):

$$C_{ij} = 62.35 + 3.47L_i + \log_{10}\left\{\frac{D_{ij}^{23.34}\cdot(T_i+T_j)^{17.06}}{(T_i\cdot T_j)^{12.68}\cdot(T_j)^{2.16}}\right\} + e_{ij} \tag{2}$$

where $e_{ij} \sim N(0,49.42)$.

Between layers II ($D_{ij} > 3$):

$$C_{ij} = 59.97 + 7.53L_i + \log_{10}\left\{\frac{D_{ij}^{40.49}\cdot(|T_i-T_j|)^{1.96}}{(T_i+T_j)^{17.30}\cdot T_j^{5.33}}\right\} + e_{ij} \tag{3}$$

where $e_{ij} \sim N(0,27.49)$.

It may be pointed out that $j < i$ and since the pairs are numbered from the center outward, j is in the outer layer relative to i.

3.2 Simulation of Pair-to-Pair Crosstalk

The statistical models developed above can be used for Monte Carlo simulation of pair-to-pair crosstalk as follows:

(a) Specify D_{ij}, T_i, T_j, and L_i for the pair combination.

(b) Compute the expected crosstalk from the appropriate one of the above three fitted models (without e_{ij} term).

(c) Add e_{ij} to the expected crosstalk where $e_{ij} \sim N\left(0,\sigma_e^2\right)$, and

$$\sigma_e^2 = \begin{cases} 49.05 & \text{for within-layer} \\ 49.42 & \text{for between-layer I } (D_{ij} \leq 3) \\ 27.49 & \text{for between-layer II } (D_{ij} > 3). \end{cases}$$

We generate a random number, r_{ij}, from $N(0,1)$ through the use of a random number generator on a computer and define $e_{ij} = (r_{ij})\cdot(\sigma_e)$.

By generating 5151 pair-to-pair crosstalk "measurements" among 102 pairs in the cable we can develop a synthetic cable of a specified twist length design.

4. Exploration of New Cable Designs

4.1 Power Sum Distribution

Before exploring new cable designs for the 102-pair cable, it is necessary to assess the performance of the design used in the experimental cable. A measure of the overall crosstalk interference in a pair is called the power sum, which is defined below. Let C_{ij} = crosstalk from pair j to pair i, $j \neq i$. Then

$$P_i = \text{Power sum for pair } i$$

$$= -10 \log_{10}\left\{\sum_{j\neq i} 10^{-C_{ij}/10}\right\}$$

Unfortunately, we do not have all C_{ij} measurements for $j \neq i$. Therefore, we cannot compute all P_i's from the 2450 crosstalk measurements for the experimental cable discussed in Section 2. First we simulate the unmeasured crosstalk as described in Section 3 and then compute P_i(i=1 to 102). Figure 3 shows a plot of P_i on normal probability apper. The power sum distribution is an indication of the overall performance of the cable. For example, the worst pair has a power sum of 39 dB and the best pair has a power sum of 48 dB. The left-hand tail is the important tail of the distribution because the crosstalk requirement is stated as follows:

$$\text{Prob.}(P_i < P_0) \leq \alpha.$$

Since the simulated crosstalk and consequently simulated P_i are random variables, the plot in Figure 3 represents one realization of the power sum distribution for the design used in the experimental cable. If we make another simulation of the unmeasured crosstalk, recompute P_i and plot the resulting power sum distribution, we will get another realization of the power sum distribution. Figure 4 shows 20 realizations of this distribution (one of which is shown in Figure 3) for the experimental cable design. It can be seen that the tails of the power sum distribution are quite variable and consequently it is not efficient to assess the performance of a given design on the basis of one realization of its power sum distribution. The average of the 20 realizations displayed in Figure 4 is shown in Figure 5 along with three other similar average distributions. It can be seen that the four average distributions are quite close to each other. Therefore, we can use the average of 20 realizations of power sum distributions for comparing one cable design with another cable design.

4.2 Validation of Simulation

The simulation of power sums discussed above is based on the statistical models developed in Section 3 which were judged to be quite useful in describing pair-to-pair crosstalk. Can we investigate the validity of the simulation of power sums more directly? Figure 6 shows four realizations of the average power sum distribution for each of the following two cases:

(i) When only unmeasured pair-to-pair crosstalk is simulated from the fitted regression models (also shown in Figure 5), and

(ii) When all pair-to-pair crosstalk are simulated from the fitted regression models.

The above two sets of distributions have a considerable amount of overlap which indicates that the power sum distribution is not affected if we replace available measurements of pair-to-pair crosstalk by corresponding values simulated from the fitted models. In other words, the simulation of the power sum distribution appears to be valid for practical purposes.

4.3 New Cable Designs

As discussed above, the statistical models fitted to measurements of crosstalk in the experimental cable lead to a proper simulation of pair-to-pair crosstalk and power sum distribution for this cable design. In this section we will assume that the fitted models are also valid for new (i.e., different from that in the experimental cable) 102-pair layer type cable designs. To minimize the lack of applicability of the fitted models, we will not consider twist lengths outside the range of those of the experimental cable.

There are 26 distinct twist lengths in the experimental cable. Since it is expensive to keep an inventory of a large number of distinct twist lengths, the cable manufacturing organization would prefer to use fewer twist lengths if crosstalk can be kept within specified limits. The exploration of new cable designs is discussed below.

First, let us consider the "tuning up" of the design used in the experimental cable. What changes in twist length assignment will improve the power sum distribution? To help answer this question let us examine Figure 7 which shows a plot of the average (of 20 "synthetic" cables) power sum for each of the 102 pairs. It can be seen that the worst power sums correspond to the pairs in layer 4. Reviewing Table 1, Figure 1 and Model (1) given earlier, the following facts may explain why the pairs in layer 4 have the worst power sums:

(i) Layer 4 pairs have the largest number of neighbors in adjacent layers.

(ii) Layer 4 pairs have long twist lengths which result in large within-layer crosstalk.

We cannot change the number of neighbors in adjacent layers for pairs in Layer 4. However, we can assign short twist lengths to Layer 4, thereby improving crosstalk within this layer. This was done by selecting short twist lengths for layers 2, 4 and 6 and long twist lengths for layers 1, 3 and 5. In addition, the number of distinct twist lengths in layer 2 was reduced from 9 to 5 and the longest twist length (7.14") was eliminated. The resulting design, labeled as Design (1) (see Table 4), improved the worst average power sums by 1 dB as shown in Figure 8 and Table 5. The number of distinct twist lengths has been reduced from 26 for the original design to 23 for Design (1).

Next, let us consider the exploration of designs with considerably fewer distinct twist lengths. By assigning the same twist lengths to alternate layers, we can do with 10 distinct twist lengths. Unlike the original design and Design (1), stranding lays of alternate layers are made different to make "effective twist lengths" different for these layers. Among the 10-twist length designs considered, Design (2) (see Table 4) was found to be the best. The average power sum distribution corresponding to Design (2) was still better than that for the original design with 26 distinct twist lengths (Table 5).

TABLE 4

Twist Length Assignments

	Original Design		Design (1)		Design (2)	
Layer No.	Range of TL	Str.	Range of TL	Str.	Range of TL	Str.
1	0.90-1.39	24R	2.96-6.25	24L	1.18-5.46	25L
2	1.77-6.25	24L	1.07-2.17	24R	0.90-4.36	35R
3	1.07-1.48	24R	2.43-5.46	24L	1.18-5.46	35L
4	2.50-7.14	24L	0.90-2.05	24R	0.90-4.36	20R
5	0.90-1.77	24R	2.30-4.90	24L	1.18-5.46	25L
6	2.05-6.25	24L	1.07-2.17	24R	0.90-4.36	35R
No. of Dist. Twists	26		23		10	

5. Summary

The crosstalk measurements performed on the experimental cable were chosen to provide estimates of inherent variability due to the measurement and manufacturing processes as well as to provide sufficient data for fitting regression models. Choice of regression variables was aided by plotting crosstalk and residual crosstalk against design variables. Resulting fitted models have a small lack of fit and were judged adequate for describing pair-to-pair crosstalk.

These fitted models and an appropriate random component of variability were used to simulate pair-to-pair crosstalk. Crosstalk in each pair from the remaining 101 pairs were "added" to get an overall index of interference called the power sum. The distribution of power sums for all pairs of a cable is a measure of the performance of the cable design which it represents. The power sum distribution was found not to be sensitive to replacement of measured crosstalk by simulated crosstalk, which fact supported the validity of the Monte Carlo simulation.

New designs have been developed for the 102-pair layer type cable by building and analyzing "synthetic" cables. Two of these are reported herein. Recall that there were 26 distinct twist lengths in the experimental cable. The first of the two new designs uses 23 distinct twist lengths and yields an improvement of 1 dB in the worst power sums, an improvement of engineering significance despite appearing to be small. The development of the second design

TABLE 5

Simulated Power Sum Distributions
For Several Designs

Ordered Power Sum No.	Simulated Power Sum (dB) For Length 1000 Ft.		
	Original Design	Design (1)	Design (2)
1	36.20	37.22	36.90
2	36.70	37.67	37.37
3	37.66	38.42	38.04
4	38.05	38.92	38.39
5	38.54	39.24	38.66
.	.	.	.
.	.	.	.
10	39.75	40.02	39.55
.	.	.	.
.	.	.	.
20	40.97	40.91	40.77
.	.	.	.
.	.	.	.
30	41.69	41.63	41.43
.	.	.	.
.	.	.	.
50	42.83	42.63	42.69
.	.	.	.
.	.	.	.
102	49.90	48.41	48.96

indicates that it may be possible to reduce the number of distinct twist lengths from 26 to 10 without degrading crosstalk power sums for the worst pairs. Such a reduction holds promise for the cable manufacturing organization for reasons of cost.

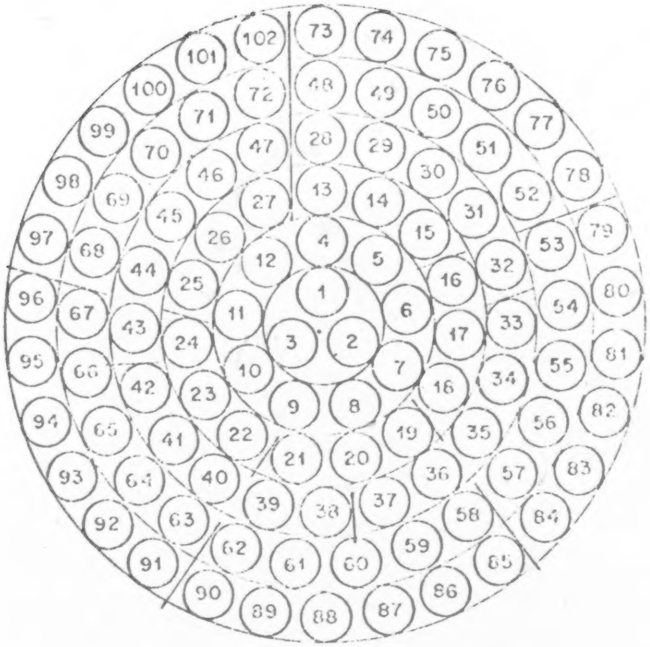


FIGURE 1 CONFIGURATION OF 102-PAIR LAYER CONSTRUCTION CABLE

NOTES 1. In each layer there is a group of distinct twist lengths. Let T_i = twist length of pair i. Then in layer 3: $(T_{13},T_{14},T_{15}) = (T_{16},T_{17},T_{18}) = \ldots = (T_{25},T_{26},T_{27}) = (1.1'',1.2'',1.5'')$.

2. Pair combinations (13,14) and (16,17) are not considered distinct pair combinations. Similarly, (15,31) and (18,36) are not considered distinct pair combinations, whereas (15,16) and (16,17) are distinct pair combinations.

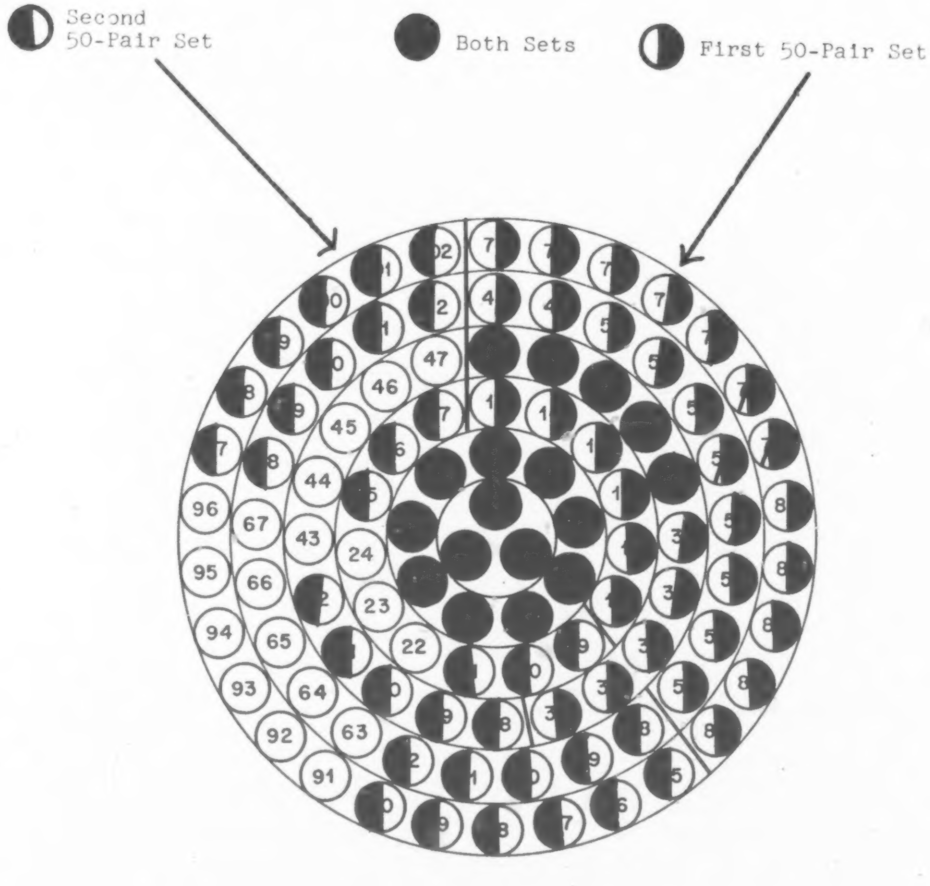

Second 50-Pair Set
Both Sets
First 50-Pair Set
47
46
45
44
43
24
23
22
96
95
94
93
92
91
67
66
65
64
63

FIGURE 2 SELECTION OF TWO 50-PAIR SETS

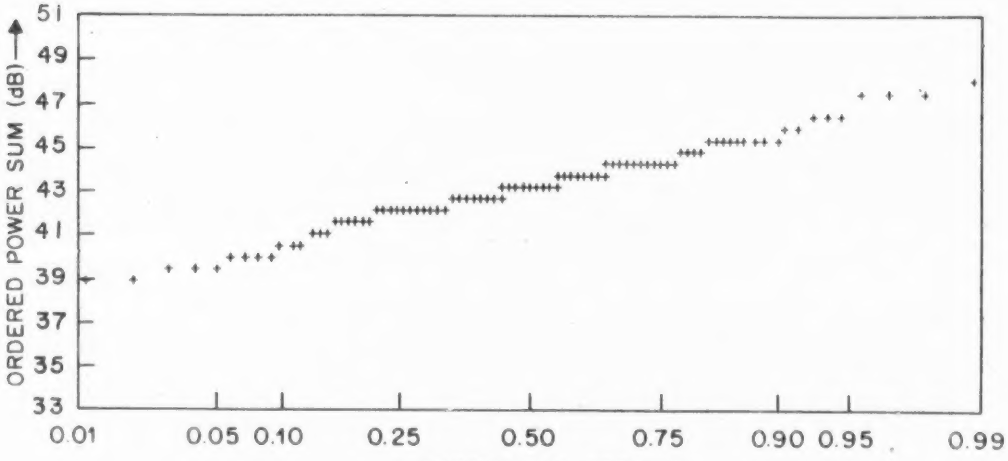


FIGURE 3 A SIMULATED POWER SUM DISTRIBUTION

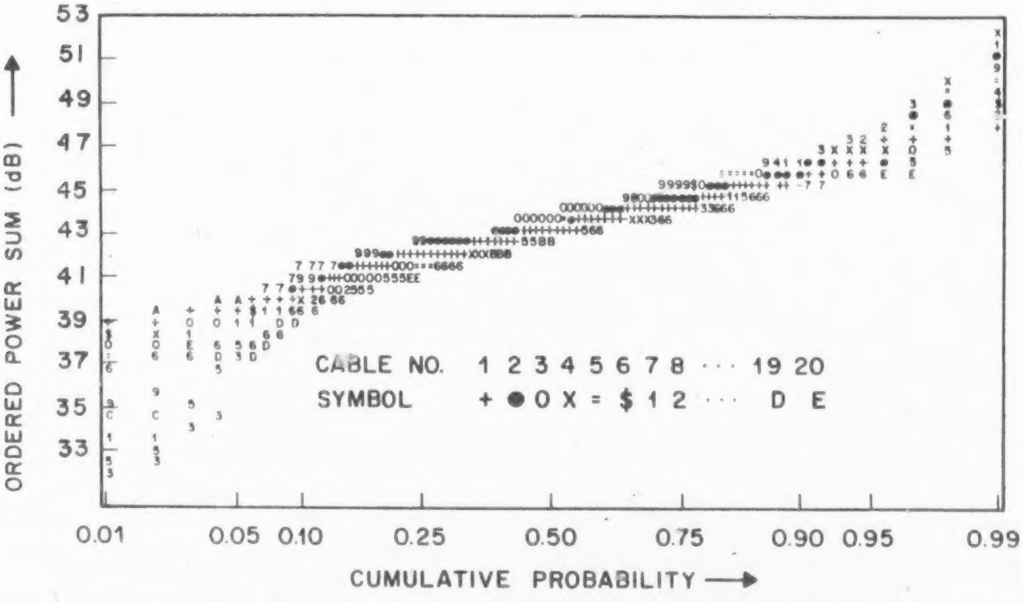


FIGURE 4 TWENTY REALIZATIONS OF THE POWER SUM DISTRIBUTION

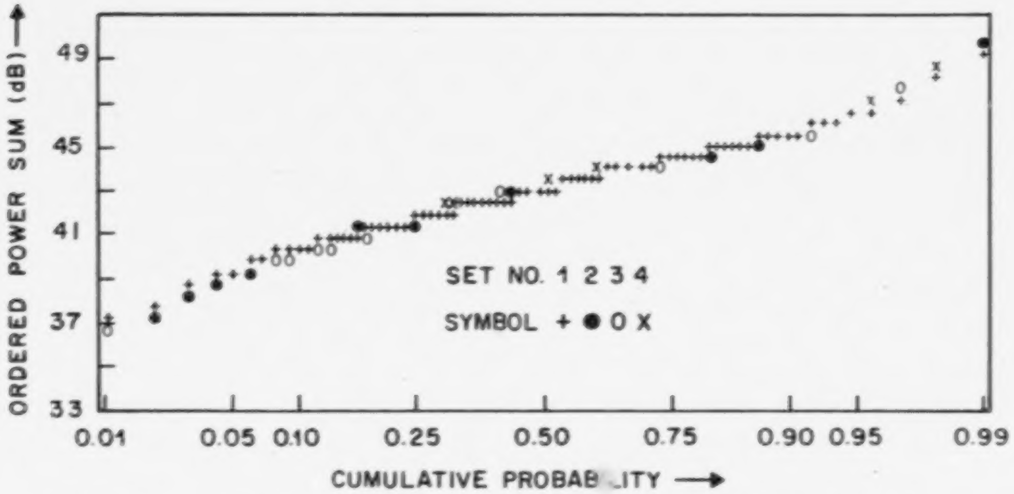


FIGURE 5 AVERAGES OF TWENTY REALIZATIONS OF THE POWER SUM DISTRIBUTION

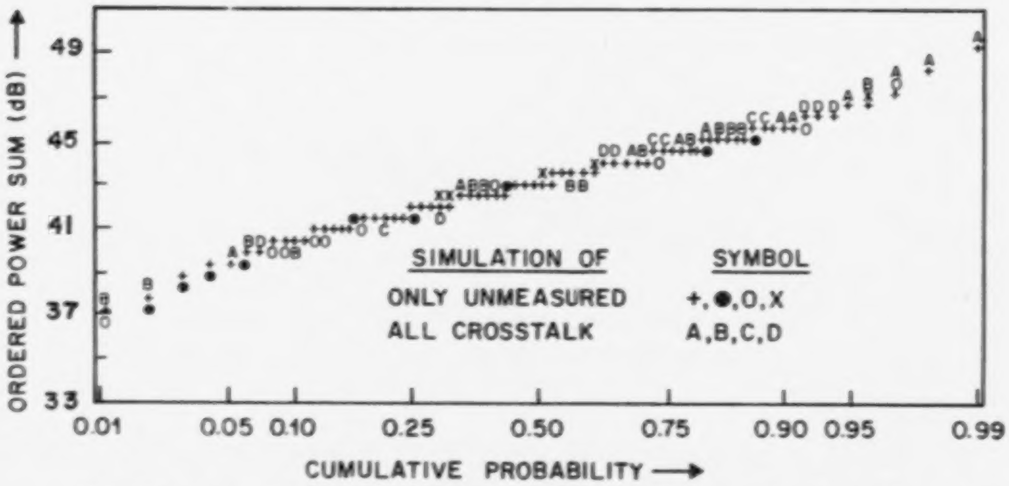


FIGURE 6 AVERAGE POWER SUM DISTRIBUTIONS WHEN ALL (OR ONLY UNMEASURED) CROSSTALK IS SIMULATED

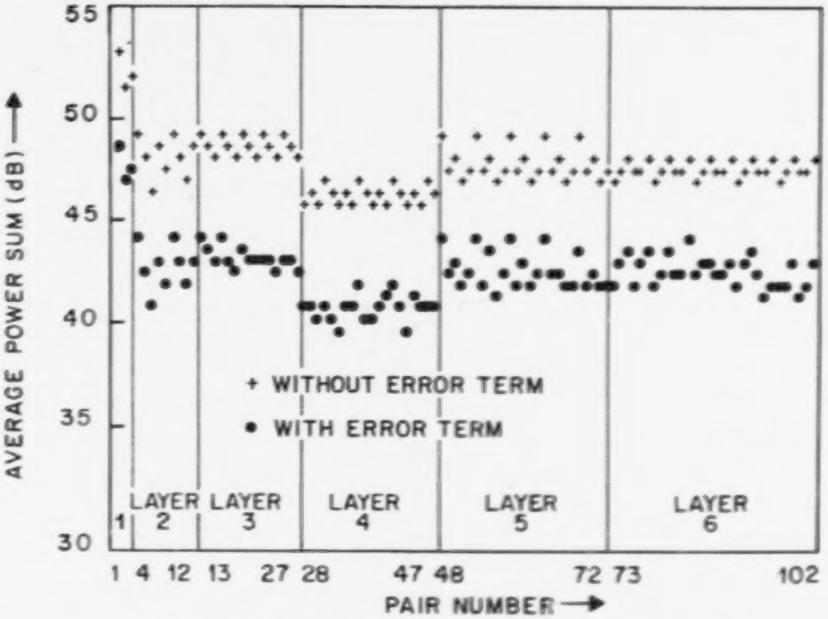


FIGURE 7 AVERAGE POWER SUM BY PAIR NUMBER

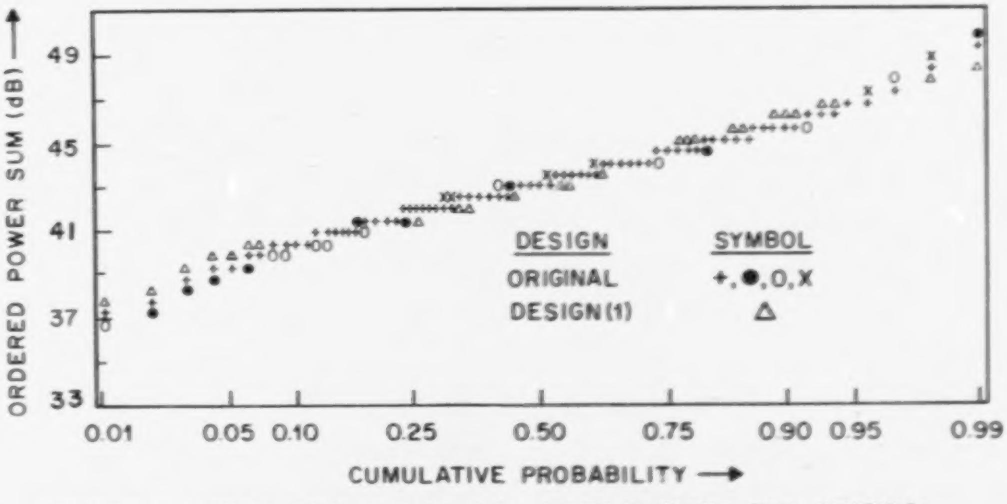


FIGURE 8 SIMULATED POWER SUM DISTRIBUTIONS FOR ORIGINAL AND NEW DESIGNS

INCORPORATION OF FALSE ALARMS IN SIMULATIONS OF ELECTRONIC RECEIVERS

V. P. Sobczynski
C. J. Pearson

SYCOM, Inc.

Abstract

A simple method for simulating false alarms in simulations of electronic receivers is developed. False alarms are an important part of electronic receivers and previously their generation was computationally complex because of their random nature and very small probability of occurrence. The approach derives the probability density function of the time between false alarms and shows how, for typical values of false alarm probability, the density function may be approximated by a discrete uniform density. Procedures are outlined for simulating false alarms both in the absence and presence of jamming. Also, a method is shown for representing the false alarm as having a finite width commensurate with the bandwidth of the receiver. A rapid computer algorithm for obtaining the amplitude of a false alarm is derived.

In digital time-ordered simulation of electronic receivers, it may be desirable to simulate detection threshold crossings due to false alarms randomly throughout the entire run. If a receiver has an ideal video bandwidth B, then the rate of occurrence of independent noise samples is 2B, corresponding to average time intervals of 1/(2B). An immediately obvious way to generate false alarms is to pick a random number from an appropriate distribution every 1/(2B) seconds. In practice, this can lead to an excessive number of computations and an unacceptably inefficient code. For example, if 2B = 20MHz, a draw must be made every

50 nanoseconds (simulation time). The purpose of this paper is to describe an alternate procedure which is generally much more efficient.

Time of Occurrence of False Alarm

The approach proposed here employs the probability density function of the time between false alarms. Starting at time zero, a draw is made from an appropriate density function to determine the time of the first false alarm. At the time determined in the first draw, a false alarm is generated and a draw is made. This procedure is continued through the entire run. The derivation of the procedure is as follows:

Let p = probability of false alarm during single time increment

and
$$q = 1 - p. \tag{1}$$

Then if a trial is considered to have occurred at each time slot (every 50 nanoseconds if 2B = 20MHz), the probability of occurrence of a false alarm on the n^{th} trial is the following for several values of n:

$$\begin{aligned} pr\{n = 1\} &= p, \\ pr\{n = 2\} &= pq, \\ pr\{n = 3\} &= pq^2, \\ pr\{n = 4\} &= pq^3, \end{aligned} \tag{2}$$

and, in general,

$$pr\{n = N\} = pq^{N-1}. \tag{3}$$

That is, the probability that the next false alarm will be generated at the n^{th} time slot is pq^{n-1}. It may be easily checked that this is a probability density function by noting that

$$\sum_{i=0}^{\infty} pq^i = 1, \tag{4}$$

by the formula for the sum of an infinite geometric progression.

It can be shown that if one specifies a percentage of the total number of time slots, then the geometric distribution may be approximated by a uniform density from zero to approximately twice the average value. Since the density function actually extends over an infinite number of time slots, it is not possible to include all of them. If it is assumed that the last 10% of the density function may be disregarded, then the point in question becomes how many time slots to include to get 90% of all possible values of n. The formula for the sum of N terms of a geometric progression is:

$$S_n = a\left(\frac{r^n - 1}{r - 1}\right), \tag{5}$$

where a = first term, and r = ratio between successive terms. Substituting values, we find

$$\sum_{n=0}^{N} pq_n = p\left(\frac{q^N - 1}{q - 1}\right). \tag{6}$$

The average value $\bar{n}$ of the random number n may be obtained without difficulty as follows. Dividing the above by p yields:

$$\sum_{n=0}^{N} q^n = \frac{q^N - 1}{q - 1}. \tag{7}$$

Taking the limit as $N \longrightarrow \infty$ gives

$$\sum_{n=0}^{\infty} q^n = \frac{1}{1 - q}, \tag{8}$$

which yields, upon differentiating with respect to q and multiplying by n and q,

$$\sum_{n=0}^{\infty} npq^{n} \equiv \overline{n} = q/p \, . \qquad (9)$$

The value of N, which includes 90% of the range of n, may be found very easily. Noting that

$$p + q = 1, \qquad (10)$$

we obtain

$$\sum_{n=0}^{N_{90}} pq^{n} = .9 = 1 - q^{N_{90}} \, . \qquad (11)$$

This gives

$$.1 = q^{N_{90}} \, . \qquad (12)$$

Solving by logarithms yields

$$N_{90} = \frac{\ln(.1)}{\ln(q)} \, . \qquad (13)$$

In general

$$N_{F} = \frac{\ln(1 - F)}{\ln q} \, , \qquad (14)$$

where F = the fractional part of the density which is retained. We can now obtain the ratio $N_{90}/\overline{n}$:

$$\frac{N_{90}}{\overline{n}} = \frac{p}{q} \, \frac{\ln(.1)}{\ln(q)} = -2.3 \, \frac{p}{q} \, \frac{1}{\ln(1 - p)} \, . \qquad (15)$$

Using the Taylor series expansion for natural log and considering that p is small yields

$$\approx -2.3 \, \frac{p}{q} \, \frac{1}{(-p)} = \frac{2.3}{q} \approx 2.3 \, . \qquad (16)$$

Note that this result is independent of the value of probability of false alarm p so long as p is small.

The quantity $\overline{n}$ may be easily computed from p, the probability of false alarm. As an example, consider the following. Let the video bandwidth B = 10MHz and $P_{fa} = p = 10^{-6}$. Then the average time between false alarms will be

$$T_{fa} = \frac{1}{2BP_{fa}} = .05 \text{ sec} \, . \qquad (17)$$

The time per slot is equal to 1/(2B). The time to the next false alarm is equal to n x (time per slot), and

$$\overline{n} \equiv N_{av} = \frac{T_{fa}}{\text{Time per Slot}}$$

$$= \frac{.05}{50 \times 10^{-9}} = 1 \times 10^{6} \, . \qquad (18)$$

For these particular parameters, false alarms occur on the average every 10^{6} time slots.

Clearly, 90% of all values of n is about twice the average value. The procedure can be very simple. To compute the time to a false alarm, pick a number out of a uniform distribution from zero to 2.3 times the average time between false alarms. The actual distribution is geometric. However, because p is small, the density function is very flat and 90% of all possible values are between zero and 2.3 times the average value. Therefore, it is possible to approximate the actual distribution by a uniform distribution from zero to 2.3 times the average value (for small values of probability of false alarm).

Alternately, a more accurate but less rapid method for generating deviates with a geometric distribution is given by Naylor, et al.(6) as follows:

$$g = (\ln u)/\ln q \, , \qquad (19)$$

where g and u are geometric and uniform deviates, respectively, and

$$q = 1 - p \qquad (20)$$

as before.

Threshold Detection

The method of setting the threshold in the absence of jamming noise will be described next. The receiver consists of hardware, active and passive devices such as filters, a preamplifier, IF amplifiers followed by a detector, and possibly an amplifier. The noise generated by the receiver (given by the receiver noise figure) can be modeled as though it were generated by a resistive element in the amplifier front end of the receiver. The overall noise voltage generated by the receiver may be considered to be Gaussian with negligible error if the gain of the preamplifier is sufficiently high (e.g., 20db or greater). The variance of this Gaussian density is

$$\sigma_T^2 = kT_eB_e \tag{21}$$

where k = Boltzmann's constant, T_e = effective noise temperature of the receiver, and B_e = effective two-sided video bandwidth. The effective noise temperature of the receiver is (F-1) 290°K, where F is the noise figure (not in db). Once the average thermal noise power is known, the threshold SNR can be determined from the required probability of detection, average time between false alarms and effective video bandwidth of the receiver. Then the threshold power can be determined by adding the required signal to noise ratio (db) to the average thermal noise power (dbm). Reference to the actual system configuration enables one to determine a threshold voltage V_T valid for thermal noise alone. V_T is an input constant for the entire run.

This voltage V_T corresponds to a threshold power P_T into the receiver. There exists an input voltage V_T' which corresponds to the output threshold voltage V_T. (V_T' is computed in the program and remains constant). Consider the Gaussian probability density of the amplitude of the noise generated by the receiver. V_T' is the value of the amplitude at the input that corresponds to the threshold voltage V_T. If P_{fa} is the predetermined probability of false alarm in presence of receiver noise only, then:

$$P_{fa} = 2\int_{V_T'}^{\infty} \frac{1}{\sqrt{2\pi\sigma_T^2}} \exp\left[-\frac{x^2}{2\sigma_T^2}\right] dx\,. \tag{22}$$

It is necessary to solve for V_T'. If one makes the substitution

$$t^2 = \frac{x^2}{2\sigma_T^2}\,, \tag{23}$$

then

$$dx = \sqrt{2}\,\sigma_T dt\,, \tag{24}$$

and

$$P_{fa} = \frac{2}{\sqrt{\pi}} \int_{\frac{V_T'}{\sqrt{2}\,\sigma_T}}^{\infty} e^{-t^2} dt\,, \tag{25}$$

or

$$P_{fa} = \operatorname{erfc}\left(\frac{V_T'}{\sqrt{2}\,\sigma_T}\right) \tag{26}$$

and

$$V_T' = \sqrt{2}\,\sigma_T \operatorname{erfcin} P_{fa}\,, \tag{27}$$

where erfc represents the familiar error function complement

$$\operatorname{erfc}(x) = \left(\frac{2}{\sqrt{\pi}}\right)\int_x^{\infty} e^{-t^2} dt \tag{28}$$

and erfcin is its inverse such that if

$$\mathrm{erfc}(x) = w \,, \tag{29}$$

then

$$\mathrm{erfcin}(w) = x \,. \tag{30}$$

This expression for V_T' is programmed to find V_T' from P_{fa} using a stored table of error function complement inverse, which will be discussed below.

False Alarm Probability with Varying External Noise Level

If there is jamming noise with power σ_j^2, the probability of false alarm p changes because the total noise power

$$\sigma^2 = \sigma_T^2 + \sigma_j^2 \tag{31}$$

is different. The new probability of false alarm is given by:

$$p_1 = 2\int_{V_T'}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2}{2\sigma^2}\right] dx = \mathrm{erfc}\left(\frac{V_T'}{2\,\sigma}\right). \tag{32}$$

It is this quantity which must be used in the geometric distribution of the number of time slots to the next false alarm. Since p_1 may not be extremely small, the actual distribution rather than the approximation may be preferable.

Time of Occurrence of False Alarms with Noise Jamming

The procedure to be used for generating false alarms in the event of jamming is as follows. When the jamming signal is received, the false alarm which has been scheduled to occur is dropped. A new probability of false alarm (p_1) calculation is made. This procedure for calculating false alarms is continued for the duration of the jamming. When the jamming ends, any predicted false alarm due to jamming that has not yet occurred is dropped and the nonjamming value of p is used again. System operation then reverts to normal.

The procedure being described here normally lends itself very well to event-controlled as well as fixed time frame dynamic simulation designs. However, one must deal with the case in which changes in noise level occur between false alarms. It is suggested merely that a significant change in noise level be cause for the recomputation of the next false alarm time. Another uniform random draw may also be used to account for the fact that the previous false alarm probably did not occur at the time of the recomputation of the time interval. If changes in received noise level occur extremely rapidly or continuously in the simulation due to narrow moving beams or particular jamming techniques or noise sources such that recomputation would be too frequent, some major modification of this technique may be necessary.

Amplitude of False Alarms

The probability density function of the amplitude of the false alarm, given that a false alarm has occurred, is Gaussian, having a variance equal to the total input noise power and having a domain such that the allowed input power corresponds to a voltage amplitude output greater than V_T, where V_T if the threshold voltage. The threshold signal to noise ratio (SNR) is determined from P_{fa}, which is a simulation parameter, and the probability of detection by readily available graphs[1, 2].

If it is assumed that all noise pulse amplitudes are normally distributed, then the distribution of false alarm amplitudes $f_y(y)$ will have the shape of the normal distribution in the region $|y| \geq V_T'$, where V_T' is again the input voltage amplitude corresponding to the threshold voltage V_T, and will vanish in the region $|y| < V_T'$:

$$f_y(y)dy = \begin{cases} \frac{a}{\sqrt{2\pi}\ \sigma} \exp[-y^2/(2\sigma^2)]dy, & |y| \geq V_T', \\ 0, & |y| < V_T', \end{cases} \tag{33}$$

where σ^2 is variance of the noise amplitude and a is a normalization factor such that

$$\int_{-\infty}^{\infty} f_y(y)dy = 1 . \tag{34}$$

It is readily discovered that

$$\frac{1}{a} = \text{erfc}\left(\frac{V_T'}{a\sqrt{\pi}}\right). \tag{35}$$

The generation of pseudorandom deviates with this distribution may be accomplished, in principle, merely by using any of the usual methods for generating Gaussian deviates[4] and rejecting those which do not exceed the detection threshold. For cases in which the ratio of noise level to detection threshold is even moderately large, however, this will result in the rejection of a very large fraction of the deviates which are generated and a correspondingly large expense in computation time. We therefore suggest that the inverse method[5] be used with tabulated functions. This method, to be described below, has the principal advantage that for every uniform deviate drawn an acceptable normal deviate may be computed by a fixed-length algorithm. If equal-interval function tables may be used the method is also quite rapid. Accuracy depends on the statistical accuracy of the uniform pseudorandom deviate generator and on the granularity and accuracy of the functional tables employed.

The inverse method of generating pseudorandom deviates y with the density function $f_y(y)dy$ consists essentially in the statement that if u is a uniform pseudorandom deviate on the interval [0, 1], then y will have the required density if y and u are related according to

$$u = F(y), \tag{36}$$

where F represents the cumulative probability distribution function of the variable y:

$$F(y) = \int_{-\infty}^{y} f_y(t)dt . \tag{37}$$

In other words, the technique consists of drawing u from the uniform density and then solving for y, the random pulse amplitude.

In particular, for the case of interest here,

$$F(y) = \begin{cases} \frac{a}{2} \text{erfc}\left(\frac{|y|}{\sigma\sqrt{2}}\right), & y \leq V_T', \\ 1/2, & -V_T' \leq y \leq V_T', \\ 1 - \frac{a}{2} \text{erfc}\left(\frac{y}{\sigma\sqrt{2}}\right), & y > V_T'. \end{cases} \tag{38}$$

This may be inverted, in principle, in the form:

$$y = \begin{cases} -\sqrt{2}\ \sigma\ \mathrm{erfcin}(2u/a), & 0 \le u \le 1/2\ , \\ \sqrt{2}\ \sigma\ \mathrm{erfcin}[2(1-u)/a], & 1/2 \le u \le 1\ , \end{cases} \qquad (39)$$

where erfcin(x) again represents the inverse error function complement, a single valued, monotonic function for real arguments.

There remains the problem of developing a rapid computational algorithm for obtaining the error function complement (required for the computation of the normalization factor a) and its inverse. (Library routines are assumed to be available for generating the uniform pseudorandom deviates u.)

The methods generally used for computing the error function complement employ the Taylor series expansion or a rational polynomial approximation for this function in the central region and use the asymptotic form for arguments larger in magnitude than about 3. Such expansions have been obtained for the inverse function, or values of the inverse function may be obtained from the direct function by means of Newton's or some similar method of successive approximations. Since speed is presumably more significant than accuracy in simulating false alarms, it is suggested that the two required functions be tabulated and a table look-up routine used.

The value of a (note: $a = p_1$) must be recomputed only when the receiver noise level or the detection threshold changes (cf. Equation 32). This generally occurs somewhat less frequently than false alarms, which require the computation of the inverse function. Therefore, a table of erfcin(x) at equal intervals of x is recommended. This same table then may be used to find erfc(w) by means of a table look-up routine for unequal intervals. It is only necessary to interchange the roles of dependent and independent tabular variables. These table look-up techniques will not be discussed here.

Because detection threshold are usually well out in the tails of the noise pulse amplitude distribution, values of the inverse error function complement for arguments near 1.0 (i.e., in the central region) are not used. In particular, it may readily be shown that arguments greater than the receiver-noise-only false alarm probability P_{fa} are not required. It is then convenient to tabulate erfcin(x) versus ln x (or even $\log_{10} x$) to obtain the proper granularity with a table of equal intervals.

Table I is an example developed for just this purpose. To accomodate both central and asymptotic regions, Table I is divided into two overlapping portions, both tabulated at equal intervals of the independent variable. If $x \ge 0.34$ the central portion is used, while if $x < 0.34$ the asymptotic portion must be used. In simulating false alarms under conditions of varying input noise levels both erfcin and erfc functions are required. Nevertheless, in a large simulation where core locations are dear, careful programming of table look-up procedures makes it possible to avoid storing tabular values of the independent variable (x). Only three of the following four parameters are required to specify each part of the table of independent variables: first value, last value, interval size, number of intervals. At the same time, further computation of erfc functions can be avoided. Linear interpolation is normally entirely adequate for the purpose of generating these random numbers.

It may be of interest to consider the range of false alarm amplitudes which the table shown will allow. If the largest value of erfcin tabulated is $1/0.1898 \approx 5.27$, it follows that the largest possible pulse amplitude which will be generated is $(5.27)\sqrt{2}\ \sigma_T = 7.4\ \sigma_T$. (If the limiting values (0 or 1) of the uniform random number are drawn, they must be rejected, since pulse amplitudes of $\pm\infty$ would result.)

Effective Pulse Width of False Alarms

The next question that arises is, "What pulse width should be assigned to a false alarm?" A false alarm is generated by random thermal fluctuations in the front end of the receiver. Before it gets to the output amplifier, a false alarm of power A may be represented by an impulse function $\sqrt{A}\ \delta(t)$, then the output of the filter Y(t) may be determined by the convolution integral:

$$Y(t) = \int_{-\infty}^{\infty} \sqrt{A}\ \delta(t - \tau) h(\tau)\, d\tau\ , \qquad (40)$$

where h(t) = the time response of the linear filter. If the filter is a single pole low pass filter, then

$$y(t) = \frac{\sqrt{A}}{RC} \int_{-\infty}^{\infty} \delta(t - \tau)\, e^{-\tau/RC} d\tau \qquad (41)$$

$$= \frac{\sqrt{A}}{RC}\, e^{-t/RC} \qquad (42)$$

is the voltage output of the filter.

Because of the finite bandwidth of the receiver being simulated, it is desired to represent the false alarm as a rectangular pulse with an "effective" pulse width. The procedure used will be to calculate the energy in the exponential pulse which is the output of the filter and assume that the false alarm is a rectangular pulse of the same energy. The energy in the pulse y(t) is

$$\int_0^{\infty} y^2(t) dt\ . \qquad (43)$$

The energy in the output pulse is then

$$\int_0^{\infty} y^2(t) dt = \frac{A}{(RC)^2} \int_0^{\infty} \exp[-2t/RC] dt \qquad (44)$$

$$= \frac{A}{2RC}\ . \qquad (45)$$

An equal energy rectangular pulse of width T would have a voltage amplitude of $\sqrt{A/(2RCT)}$. If a time T = RC is chosen, then the amplitude of the pulse is $(\sqrt{A/2})/(RC)$. The false alarm is then modeled as a pulse of amplitude $(\sqrt{A/2})/(RC)$ and width RC, where RC is equal to $1/(2\pi B_d)$, and B_d is the 3db bandwidth of the detector. If at the time of occurrence of a false alarm an input signal is present, then the amplitudes of the signal input and false alarm sum together.

ln x	1/erfcin(x)
-1.0	1.5705586
-1.5	1.1608596
-2.0	0.94698245
-2.5	0.81336734
-3.0	0.72087959
-3.5	0.65246551
-4.0	0.59945271
-4.5	0.55694114
-5.0	0.52194136
-5.5	0.49251962
-6.0	0.46736597
-6.5	0.44555928
-7.0	0.42643151
-7.5	0.40948547
-8.0	0.39434236
-8.5	0.38071033
-9.0	0.36835655
-9.5	0.35709646
-10.0	0.34678015
-10.5	0.33728465
-11.0	0.32850803
-11.5	0.32036523
-12.0	0.31278465
-12.5	0.30570521
-13.0	0.29907443
-13.5	0.29284743
-14.0	0.28698569
-14.5	0.28145508
-15.0	0.27622580
-15.5	0.27127180
-16.0	0.26657003
-16.5	0.26210000
-17.0	0.25784345
-17.5	0.25378410
-18.0	0.24990731
-18.5	0.24619993
-19.0	0.24265009
-19.5	0.23924703
-20.0	0.23598100
-20.5	0.23284312
-21.0	0.22982530
-21.5	0.22692014
-22.0	0.22412085
-22.5	0.22142120
-23.0	0.21991547
-23.5	0.21629837
-24.0	0.21386504
-24.5	0.21151096
-25.0	0.20923195
-25.5	0.20702413
-26.0	0.20488390
-26.5	0.20280790
-27.0	0.20079300
-27.5	0.19883629
-28.0	0.19693503
-28.5	0.19508668
-29.0	0.19328883
-29.5	0.19153924
-30.0	0.18983581

Table I. Inverse Error Function Complement (erfcin)

REFERENCES

1) Skolnik, M. I., Introduction to Radar Systems, McGraw-Hill Book Co., New York, 1962.

2) D'Franco, J. V., and W. L. Rubin, Radar Detection, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

3) Abramowitz, M. and I. A. Stegun (Editors), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards, Washington, D.C., 1964, Sec. 7.1.2.

4) Ibid., pp. 952 et seq.

5) Ibid, p. 950.

6) Naylor, Thomas H., Joseph L. Balintfy, Donald S. Burdick, and Kong Chu, Computer Simulation Techniques, John Wiley and Sons, Inc., New York, 1968.

Tutorial 5: AN INTRODUCTION TO SIMSCRIPT

Chairman: R. Paul Wyman, Pennsylvania State University

SIMSCRIPT is a computer language designed for discrete-state simulation. It possesses all the capabilities of FORTRAN plus several powerful features such as dynamic storage allocation and list processing. Features designed specifically for simulation include the ability to describe systems in terms of entities (which are "blocks" of attributes), sets (which are groups of entities) and events (which are "scheduled" subroutines). The benefits of and disadvantages of SIMSCRIPT are briefly compared to GPSS and FORTRAN. The concepts of SIMSCRIPT are illustrated in detail using a plague epidemic example. Other examples will be briefly presented to illustrate the versatility of the language.

PAPER FAIR

Chairman - Andrew Kolterman, IBM Corporation

Many authors submitted abstracts of papers they wished scheduled for presentation in the informal atmosphere of a paper fair. In a fair, full length copies of each paper are available from authors who are assigned a booth at multiple times when they make informal presentations to the interested parties who assemble at the designated times and places. This method of presentation generates contributions from listeners who are free of the inhibitions which are often present in the large group formal presentations of the regular sessions. Abstracts of the 28 papers which were submitted to the paper fair follow.

THE DAUGHTER OF CELIA,
THE FRENCH FLAG
AND
THE FIRING SQUAD

by

G.T. HERMAN & W.H. LIU
Department of Computer Science,
State University of New York at Buffalo,
Amherst, New York

In earlier work (1, 2) we have reported on a program, called CELIA, for simulating the behavior of linear iterative arrays of cells. The action of a cell is influenced by both its neighbors. Our main application area was biology, where the program can be used to test hypotheses about the developmental rules for organisms.

More recently we have been applying our program to test proposed solutions to some fairly complicated biologically based problems. We have investigated (3) whether one can achieve regulative global polarity in organisms without polarity in individual cells, by solving the French flag problem of Wolpert (4) using only symmetric elements. In our attempted simulation of pigmentation patterns on the shells of sea-snails, we have found it necessary to find a solution to a generalized version of the firing squad synchronization problem (3), in which the firing squad is growing while it is trying to synchronize itself.

We found that our original program CELIA was somewhat awkward for simulating such complicated situations. A new program has been devised where the state of the individual cells is described by an array of attributes, such that each attribute itself may be a list structure.

The present paper reports on these changes and explains how they turn out to be useful in the applications mentioned above.

REFERENCES

(1) BAKER, R. & HERMAN, G.T., CELIA - A Cellular Linear Iterative Array simulator, Proceedings of the Fourth Conference on Applications of Simulation (1970), 64-73.

(2) BAKER, R. & HERMAN, G.T., Simulation of organisms using a developmental model, International Journal of Bio-Medical Computing 3 (1972), 201-215.

(3) HERMAN, G.T., Models for cellular interaction in development without polarity of individual cells, International Journal of Systems Sciences 3 (1972), 149-175.

(4) WOLPERT, L., The French flag problem: a contribution to the discussion on pattern development and regulation, Towards a Theoretical Biology, v.2 (Ed. C. Waddington, Pub.: Edinburgh University Press, 1968), 125-133.

"AN APPLICATION OF SIMULATION TO DEBUGGING AND MAINTAINING A COMPUTER NETWORK SYSTEM"*

by

M.W. Collins

and

D.G. Harder

University of California

Los Alamos Scienticic Laboratory

The Computing Division of the Los Alamos Scientific Laboratory is implementing a system which will link the Laboratory's large general purpose computers (3 - CDC 6600's and 2 - CDC 7600's) to a common data base (10^{12} bits on-line), and to a network of keyboard and computer based terminals. This paper discusses the use of simulation in designing, implementing, testing and maintaining the system software for the Front End Machine which is the center of this system.

An Experimental Evaluation of Monte Carlo Simulation in MIS Project Decisions

by

John B. Wallace, Jr.

University of Florida

The paper describes the methodology and results of an experimental application of Monte Carlo Risk Analysis to the process of selecting and controlling Management Information System development projects. The investigator tested hypotheses concerning tradeoffs among sophistication of the risk analysis simulation programs, alternative techniques for estimating benefits of an MIS, and managerial acceptance of the results of simulation programs as guides to project decisions.

The experiment was conducted using four different computer programs (two Monte Carlo and two non Monte Carlo) and three benefit-estimation paradigms (incorporating the managerial viewpoint, the management science viewpoint, and the opportunity-cost viewpoint, respectively) as applied to two actual MIS development projects. The data of the experiment were collected using structured interviews and taped de-briefing sessions.

Throughout the experiment, management increasingly asked for and relied on the simulation programs as decision aids. In the introductory stages, the managerial acceptance of Risk Analysis was not sensitive to the sophistication of the simulation approach. The deterministic computational packages have the advantage of manually reproducible computations and relatively inexpensive compilation and execution. The opportunity-cost paradigm found greater acceptance than the other two, but no paradigm was best in all contexts. In general the techniques tested in the experiment were cost/effective on the decisions.

ROBUSTNESS

AND

ANALYTIC COMPUTER SYSTEM MODELS

By R. Vavra and W.R. Franta

University of Minnesota

Minneapolis, Minnesota

The use of simulation studies to investigate the behavior of computer hardware/software systems is well established, although in some sense analytic models are more desirable. Simulation is used in a situation which is completely intractable to analytic machinery, or for which the essence is lost when the prerequisite abstractions and simplifying assumptions necessary to the analytic technique are made. In this paper, the term analytic model is used to refer to that set of computer system models based upon queuing theory.

Many view the operating system as a network of resources each potentially precipitating queues. Such a view allows the units passing through the network to request (employ) the devices in various combinations in a multiprogramming environment. The computing system is thus viewed as sequences of resource seizures and releases (processors, memory and input-output equipment) with appropriate holding time assumptions. In many cases, the models indicate rather clearly how various policies and variables interact to effect system performance, and the predictions of some coincide well with measurable data.

For such network models it is informative to ascertain information concerning the robustness of the model. Stated differently, we are interested in knowing to what extent the operating characteristics of the models are independent of the probability distribution and queuing discipline assumptions necessary to the analytic tractability of the model. Such investigations naturally appeal to simuation. Basically, perturbations, are made to the original distribution assumptions, and simulation experiments are conducted and results compared with those produced by the model equations. It would seem that much of this kind of work is done, although seldom reported.

This report discusses a simulation study designed to investigate robustness for a particular computer system network model briefly described as a three-stage closed queuing network. The study conducted included experiments which provided for selectable proportions of various distributions, as well as various queuing disciplines. The study extensively employed the use of antithetic variates.

A MODEL FOR SIMULATING AND EVALUATING THE RESPONSE OF A MANAGEMENT INFORMATION SYSTEM

by

Hamad Kamal Eldin
Professor, Oklahoma State University
Stillwater, Oklahoma

and

David Leon Shipman
Ph.D., Center Plans and Resources Control Office
NASA/MSFC
Huntsville, Alabama

This paper discusses the development of a model to simulate the information flow of a program management organization. The objective of the study was to evaluate the response of the information system when changes to the system were made.

The model is written in GPSS II for use on the UNIVAC 1100. It contains a different generate block for each type message entering the information system. As each message is created, it is assigned a destination, delivery duration, type of processing, and a processing duration. Assignments are made from a distribution describing that activity. Outgoing messages are created by the incoming messages and are assigned destinations and durations from corresponding distributions.

In developing the model, an information decision scheme was used. Each message is represented by a transaction and each decision maker is represented as a facility. The flow of messages through the system activate the nodes and create the output statistics.

Testing of the model was accomplished by using empirical data from a program management information decision system. During testing, it was necessary to modify the distributions several times to fit the model output to the empirical data.

After testing, the model was used to study system characteristics for different alternatives. Only the Facility and Queue standard outputs were used in the analysis. The objective of the alternative study was to determine which alternative would give maximum facility utilization without creating excessive queues. This model will assist the manager in decision making but does not provide an optimum solution.

TRACE-DRIVEN SYSTEMS MODELING

by J. F. Grant
IBM System Development Division
Endicott, N.Y.

Predicting the effect of changes made to computer operating systems is an extremely difficult task. Where possible, to a limited extent, some modeling has been done; however, because of its complexity, this modeling has been isolated to areas of very high cost or high priority real-time systems.

Trace-driven modeling is a procedure for alleviating both the high cost and difficulty of simulating changes made to computer system environments. As a result, this approach could make computer systems models an everyday tool well within the reach of the average operation.

This paper describes the attributes of the trace portion of this modeling approach and how its output might be used to drive various systems models.

ARPEGE: Simulation of an Air Pollution Crisis

M.A. Greene, A. Hochhauser, M.J. Reilly,
J. F. Sautin, A.S. Walters
ENVIRONMENTAL STUDIES INSTITUTE
Carnegie-Mellon University
Pittsburgh, Pa.

Air pollution episodes represent a major environmental problem in urban areas. The high concentration of pollution sources in such areas results in dangerously high levels of air pollution during periods of poor weather conditions. The avoidance of hazardous levels depends upon rapid and decisive actions by responsible people together with cooperation of an informed public. There is a need for educational materials which treat effectively the problem of air pollution episodes.

The Air Pollution Episode Game is an educational tool which treats the relevant aspects of urban life during a period of high air pollution. The participants assume various realistic roles and receive information and make decisions characteristic of those roles. The heart of the game is a computer simulation program which calculates meterology, emissions, air quality, adverse effects, etc.

A SYSTEMS DESIGN GAME

Norman R. Lyons
Graduate School of Business and Public Administration
Cornell University
Ithaca, New York

A major problem for a manager who used computer services is the problem of systems acquisition and design. He must decide which system to buy and how the system should be structured. This paper presents a computer game written in PL/I that enables a user to choose from among three basic computer systems with a wide variety of CPU, channel and peripheral equipment options the one that best fits the computing needs of a hyothetical organization. A simulated set of jobs is run on the configuration chosen, and throughput and equipment utilization statistics are reported.

A Deterministic Simulation Model For Scheduled Airline Fleet Maintenance

Alan J. Parker
Division of Organization and Administration
School of Business
Florida International University
Miami, Florida

The system described by the simulation is the operation and scheduled maintenance of a fleet of fifty Boeing 727 aircraft. Sixty airports are serviced with a total of 286 flights a day. The model concentrates on scheduled (periodic) maintenance set by FAA specifications.

This encompasses fifty percent of all maintenance work. The major focus of the results, at the tactical level, is on the utilization of flying hours between the various maintenance checks At the strategic level, the model can answer many questions, such as the impact of moving and closing maintenance bases and the effect of changing the number of spare aircraft.

ASSET, A Digital Computer Language for the Simulation of Communication Systems

by

R.R. Bowen
C.D. Shepard
R.V. Baser
Communications Research Centre,
Department of Communications
Ottawa

ASSET has been written to simulate time-continuous communication, control, and radar systems on the XDS Sigma 7. Both the detailed responses of such systems to specific inputs, and system performance characteristics such as signal to noise ratio, probability of error, probability of detection, mean squared error, etc. can be measured with ASSET. In general, the simulation technique is to convert the continuous-time system to a sampled-data system, and then to represent the sampled-data system blocks by ASSET statements. ASSET was designed with three goals in mind: ease of programming, measurement accuracy, and efficient Monte Carlo simulation; careful choice of measurement technique and compiler design resulted in what the authors believe to be an optimum compromise between these sometimes conflicting requirements.

SIMULATION MODEL OF A MULTIPHASIC SCREENING UNIT FOR USE BY A DEPARTMENT OF SURGERY

Frances M. Delaney, M.S.
Marilyn Oppenheim, M.S.
Martin Goldberg, M.S.
William Schumer, M.D.
A. Gerson Greenburg, M.D.

Increased emphasis has been placed on the utilization of automated multiphasic screening units as a means of contributing to the improvement of health-care delivery. This paper describes the application of a GPSS program to evaluate such a unit for use by a surgical service. By using the simulation model, statistics on the efficiency and utilization of the unit are obtained and compared with real-world data. Variations in patient scheduling and processing made within the model assist in the determination of the most effective use of fixed health-care personnel and facilities. Based upon model-generated data, predictions can be made about personnel and facility capacities required to cope with the increasing demands projected for services in the hospital. Application of a simulation model to the planning, analysis and implementation of a multiphasic screening unit can lead to economic benefits and efficient service for patients and physicians.

MEDICAL CARE SIMULATION; A STUDY UTILIZING DYNAMIC SIMULATION MODELING

S.H. Cohn and J.F. Brandejs
Department of Industrial Engineering
University of Toronto
Toronto, Ontario
Canada.

Development and implementation of a dynamic computer-aided simulation model of the Family Practice Units Network affiliated with the University of Toronto is discussed. The medical network is conceptualized in an industrial dynamics framework, as a system of interacting flows of patients, medical staff, capital assets, information, and money.

Given a patient demand for medical care, the model will process patient's allocation, hire staff, accumulate capital assets and generate expenditures.

The model is used to develop an understanding of the dynamic behaviour exhibited by the different family practice teaching units, and will predict the dynamic consequences of variations in family and community health care policies.

SIMULATING THE IMPACT OF EXPANDED DELEGATION OF DENTAL PROCEDURES

J.B. Dilworth
W.J. Pelton
O.H. Embry
G.A. Overstreet
School of Business
The University of Alabama
Birmingham, Alabama

This paper discusses a project underway at the University of Alabama in Birmingham Dental School. The study objective is to investigate the productivity of dental teams using therapists. Therapists, sometimes called technotherapists or expanded duty dental auxiliaries, are dental assistants who perform under the supervision of dentists some of the reversible procedures previously performed only by dentists. The study involved both an actual practice using varying numbers of therapists and a GPSS model which simulates practices with varying numbers of therapists. Some of the reasons are discussed for using the simulation model to confirm and expand the actual practice findings. The paper also presents the model and discusses some of its possible future applications. Productivity data from the actual practive are compared to data from simulations. Simulation data are presented which compares the services provided by a dental team using a therapist to the services provided under identical conditions when the therapist and her assistant are excluded from the team.

A PL/I Model of an Emergency Medical System

Kenneth F. Siler
Computer Methods and Information Systems
University of California
Los Angeles, California

A comprehensive PL/I simulation model has been developed for evaluating existing and proposed emergency medical systems. The model is composed of two sub-models and a unique analysis package. The first sub-model generates a "representative" stream of emergency incidents from user specified tables. This incident stream becomes the input to the second sub-model which simulates an emergency medical system responding to the incidents. An event-oriented methodology is used in the simulation. Performance of the model is validated by simulation using actual data from the San Fernando Valley area of Los Angeles.

The model is keynoted by a user orientation, flexibility, and generality which is not found in other EMS models. Al input to the model is checked for consistency to avoid erroneous computer runs. Numerous dispatch and retrieval alternatives for emergency vehicles can be tested by varying only a few parameters of the simulation. Furthermore, both simple and complex emergency medical care systems can be represented easily.

Results of the simulation using the two sub-models is analyzed by a unique PL/I program. Using the preprocessor facility of PL/I, a tailor-made analysis program is constructed. In simulations involving many variables, such as an EMS simulation, it is almost impossible to design an analysis program that satisfies all relevant research. To circumvent this problem, the analysis package allows the user to specify his desired analysis and then constructs a PL/I program to do it. Experience with the package demonstrates its usefulness and efficiency.

900

SIMULATION OF AN EPIDEMIC; DEVELOPMENT OF CONTROL STRATEGIES OF SCHISTOSOMIASIS

Keh-Lon Lee
Department of Electrical Engineering and Computer Sciences
University of California
Berkeley, California

Schistosomiasis is a vector-borne parasitic epidemic currently affecting about 250 million people and constitutes a serious public health problem in many countries. In Egypt alone, the estimated annual loss due to this disease is about $560 million dollars.

The life-cycle of schistosomes involves two incubation periods, one in human beings, another in snails. We found that in this case, it is most appropriate to use a newly developed modeling technique - the Delay-Line Model approach for the population dynamics. Our model can easily and naturally handle the time delays inherent in this system. It is easy to simulate and easy to modify on SNAP - a conversational drawing program that enables a user to create a topological network using a light pen and CRT. Data for simulation are taken from field work study results in Khuzestan, Iran.

Certain control measures are available. Such measures include chemotherapy, pesticides for vector control, environmental engineering measures, and sanitary engineering measures. However, as pointed out by expert researchers in this disease, application of a single method cannot lead to eradication, and the greatest need now appears to be the considered implementation of different combinations of control measures based upon well founded data, and their proper evaluation. Based on our model and using theoretical stability considerations as well as some heuristics derived from simulations, a set of combinations of control measures is presented.

SIMULATION ANALYSIS OF AN EMERGENCY CARE FACILITY

E.C. Garcia
W.F. Hamilton
J.W. Thomas
Department of Management   Department of Community Medicine
The Wharton School   The School of Medicine
University of Pennsylvania
Philadelphia, Pennsylvania

A GPSS model has been developed to assist in the planning and evaluation of emergency medical facilities. This paper describes the ERSIM Model and its use in the analysis of design and operating alternatives. Applications of the model to date have included analysis of triaging policies and physician staffing patterns. The results of these studies and opportunities for future applications are discussed.

AN INTERACTIVE MULTI-ITEM INVENTORY COMPUTER SIMULATION MODEL

Dr. M. Wayne Shiveley
Lehigh University
Department of Industrial Engineering
Bethlehem, Pennsylvania

A generalized inventory simulation model has been developed to establish the value of a company's inventory. This model was developed to evaluate inventories which are made up of subassemblies, assemblies, and finished goods; therefore, one component of the model is a time-sharing bill of material processor. The model accepts a finished goods forecast for spec-

ified time periods; it then predicts the net requirements for these time periods. Also the obsolete items are identified. The model can be used for production planning as well as the evaluation of on the shelf inventories. The model is interactive; it allows the user to vary inputs from a portable terminal and identify and critical parameters of the model. The model can be demonstrated to any interest group which can provide access to a standard telephone.

GWSS - A GENERALIZED WAREHOUSE SIMULATOR SYSTEM

Alvin M. Silver
Dasol Corporation
New York, New York

A generalized warehouse simulator system (GWSS) was constructed to facilitate the construction and operation of simulation models of complex warehouse systems by design engineers and operating managers. This paper presents the structure of the generalized model and the techniques used to provide extreme versatility in the warehousing system that can be modeled. The use of the generalized model is explained and its application in the construction and exercising of a simulation model for a large complex warehouse system is illustrated by an example.

A DYNAMIC CONTROL SYSTEM FOR HOSPITAL INVENTORIES

by

James D. Durham
The MEDICUS Corporation
and
Stephen D. Roberts, Ph.D.
University of Florida

This paper offers an approach to the inventory problem involving stochastic demand and stochastic lead time when only empirical distributions of these random variables are available. The proposed inventory model utilizes experimental optimization in a unique fashion to solve this problem.

Through the use of Monte Carlo simulation and a modified non-linear programming approach, an expected total cost objective function is minimized by the selection of appropriate reorder points and reorder quantities. The model is shown to be dynamic in nature and suitable for control of large inventories. The technique has been applied to several hospital inventory items and sample computations are included in the paper.

DISTRIBUTION COMBINING PROGRAM

Oldrich A. Vasicek
Wells Fargo Bank
San Francisco, California

The Distribution Combining Program is a set of algorithms that evaluates the probability distribution of the sum, difference, product, or ratio of two random variables with specified distributions. The method allows for correlation between the two input variables. The input

distributions are assumed to be from the semi-normal family (i.e., with density function that is composed of two normalized halves of normal densities with generally different variances). The resultant distribution is approximated by a distribution of the same type in order to provide subsequent use as an input in a series of operations.

The program uses the technique of calculating several moments of the output distribution from the moments of the input distributions. This method avoids using Monte Carlo simulations, or any numerical evaluation of the convolution integrals. The program can be used in risk analysis, subjective probability evaluation, sensitivity analysis and similar situations. It is particularly suitable to replace Monte Carlo methods where correlated variables are involved, or when rapid execution is desired.

MATHRISK - A MANAGEMENT TOOL FOR THE ANALYSIS OF INVESTMENT DECISIONS

Stephen L. Robinson
Mathematica, Inc.

In recent years simulation has played an increasingly more prominent role in the analysis of new investment opportunities. Numerous computer programs have been created to facilitate the simulation of cash flows created by new investments. Such programs can usually be classed as either inordinately simple to use, in which case they are often quite inflexible, or quite difficult to use, for which price the user obtains a flexible program.

This paper describes the design criteria for a dynamic software system which is not only easy to use, but flexible enough to provide the user with progressively more complex modeling capability.

MATHNET: A REPRESENTATION AND ANALYSIS TECHNIQUE FOR STOCHASTIC NETWORKS

Stephen L. Robinson

Mathematica, Inc.

The methodology presented here was developed in response to the need for a simple way to perform time-cost tradeoff analysis of research and development programs. MATHNET has been adopted by members of the managerial community as an effective tool for such analyses. Its enthusiastic reception can be directly traced to its development history.

MATHNET was originally designed as a teaching aid for a seminar on the risk analysis of R & D projects. As a result it is extremely simple to learn. Persons familiar with PERT, or other network representation schemes, have learned MATHNET in the course of very short training sessions.

The extension of MATHNET's capabilities has been dictated by situations encountered by MATHEMATICA personnel in the course of conducting risk analyses of a wide class of programs. Most R & D projects, therefore, are representable by existing MATHNET symbology. The modular design of MATHNET provides for the easy addition of symbols to represent decision types not presently incorporated in the system.

MATHNET has already proven to be of great value in the risk analysis of several large-scale research and development programs. It is anticipated that MATHNET will have an impact on the analysis of programs requiring stochastic representations similar to the impact that PERT had on programs representable by deterministic networks.

A RISK-RETURN SIMULATION MODEL OF COMMODITY MARKET HEDGING STRATEGIES

Robert E. Markland
Associate Professor-Management Science
University of Missouri - St. Louis
Robert J. Newett
Consultant-Operations Research
Ralston Purina Company

The American food processing industry is characterized by a vast array of products, which are produced in large quantities at relatively low unit costs. The pincipal component of the unit cost for these food products is their raw material (usually a basic grain commodity) constituent. Consequently, most food processing companies are greatly concerned with the prices they pay for their raw materials, and as the prices of these raw materials change, the typical food processing company's profits may be greatly affected. Since most companies prefer a steady growth rate, these raw material price fluctuations must be counter-balanced by other strategic or operating decisions. One basic set of decisions which is utilized to overcome raw materials price fluctuations involves the established commodity trading markets.

The operation of the commodity futures markets allows food processors to determine what prices they will pay for their raw materials over the production horizon. However, since a number of futures options may exist for each commodity, the inherent risk of price fluctuation remains with the processor, as some manufacturers may buy their supply of the commodity at significantly lower prices than others and reflect this difference in the price of the finished product.

The research described in this paper was conducted within the commodity market trading environment of a major American agri-business firm. The objective of the study was the development of a risk-return simulation model which could be used for testing commodity market hedging strategies in both cash and futures markets. The model was developed under general assumptions for the basic commodity, corn, and included a market price change simulation subsystem, a hedging strategy testing subsystem, and a risk-return measurement subsystem. The simulation model was tested over a multi-period time horizon for a series of commodity market hedging strategies, and extensive test results are presented.

USING THE COMPUTER TO PLAN PRODUCTION IN A FLOW SHOP

Dana B. Hopkins, Jr.
Babcock & Wilcox, Alliance, Ohio

Management of one of the Company's product lines must submit contract bids during one year for manufacture, with start and due dates, any time in the next five years. In the past, management had insufficient information as to the effects a new contract would have on their manpower requirements, resource utilization, and present contracts. The solution was to simulate production. The results of the simulation was a general schedule for production along with manpower utilization under given capacity constraints. This general schedule is not used for day-to-day scheduling of operations. Rather, it is used for "middle range planning", three months to a couple of years, where manpower levels and equipment are variables instead of constraints. Management reviews the output and makes any desired capacity and/or contract changes. A new simulation is made and the process repeats itself until management has determined what contracts to bid on and what their manpower and equipment needs will be during the "middle range".

A Simulation Study of Basic Oxygen Furnace Operations

C. Jain, Ph.D.
Assistant Professor, The Cleveland State University
Phil McDermott, Group Leader
Jones and Laughlin Steel Corporation

The study was conducted in the Basic Oxygen Furnace shop of a reputed steel company. Data were collected over extended periods to establish the statistical characteristics of daily liquid iron (hot metal) production from the blast furnaces, the life of refractory-lined vessels of the Basic Oxygen Furnace, and the heat-cycle time. This simulation generates total daily hot metal production from the specified distribution and determines the optimum combination of hot-metal-only charge (regular heat) and mixed charge (scrap and hot-metal or pre-heat) that will maximize ingot steel production within a fixed time period. The simulation takes into account the heat-cycle-time, the charge-mix, the number of heating-vessels available on the oxygen furnace, and the time required for relining of vessels. An iterative process using a modified form of the simplex algorithm with two constraints have been utilized in the simulation study for maximizing output. The output includes, on a daily basis, the hot-metal produced in tons, the number of vessels available, the number of regular-heats and pre-heats required for maximum steel production, steel production in tons, the amount to be pigged, and the amount available for the next day. The simulation provides the decision-maker in charge of the BOF operation with an operational discipline to maximize productivity and a reliable indicator to justify installation of scrap preheating facilities.

The entire simulation is written in Fortran IV, level G, and executed on an IBM 360 Model 65 computer.

AUTONETICS PLANNED PRODUCTION LINE EVALUATION SIMULATOR (APPLES)

P.J. Moore, Autonetics Division, North American Rockwell

Production line evaluation at Autonetics for the manufacture of electronic and electromechanical aerospace products, is facilitated by APPLES, a model written in GPSS/360 Version 5. The model may be applied to a large variety of production lines with differing configurations through the use of standard data forms. It is designed for use by production engineers that are unskilled in GPSS. A simulation analyst provides consulting services to the engineer as input data are prepared and assists as necessary in the experimentation. However, each user is sufficiently briefed that he is able to submit his data and conduct his study independently.

APPLES is primarily intended for specialized production lines which are to be treated as systems or subsystems in the study of their production capability. Facility dedications, manpower availabilities, work assignments and scheduling alternatives are readily evaluated wherein productivity, yield, and process times are the major stochastic variables. The basic program is easily adaptable to any unique application by the addition of subroutines that simulate the unique relationships. Thus, a single development activity has provided a generally useful tool that can quickly and directly solve ordinary problems and shortcut the development of very complex models.

The content and format of the output reports generated by APPLES are designed to relate performance reasults to production cost factors. A cost per unit index is calculated for each experimental run.

The paper describes the simulator's design, illustrates its applicability and discusses the experience achieved at Autonetics.

A PERFORMANCE EVALUATION TECHNIQUE FOR THE MEASUREMENT OF A FACILITY'S ABILITY TO PROCESS THE PROPER JOBS

J.J. Babiel and B.Z. Duhl
IBM System Development Division
Endicott, N.Y.

Most manufacturing floor control installations perform substantial analysis in choosing appropriate job sequencing algorithms with the aid of simulations, but tend to neglect further analysis required to continuously measure the degree of compliance to job sequencing rules. With the advent of on-line, real-time capabilities, priorities of jobs can change throughout the working day and frustrate the efforts of a facility in striving to "do the right jobs".

Further, different levels of work-in process were found to be major factors affecting the ability to comply with sequenced worklists.

This paper describes the various measurement schemes considered, highlighting the advantages and pitfalls of each. It also illustrates an effective reporting technique which lets management evaluate how manufacturing and/or production control areas are influencing customer serviceability.

A Directed Search Approach to Selecting a Sequencing Rule*

Dr. James C. Hershauer
College of Business Administration
Arizona State University
Tempe, Arizona
Dr. Ronald J. Ebert
School of Business
University of Washington
Seattle, Washington

A standard approach to selecting a simple sequencing rule for decentralized application throughout a job shop is developed and illustrated. Search procedures are applied to a response function which is an expectation of relevant cost per order. The cost for each order observed is a weighted combination of the multiple responses that exist in a job-shop environment. An expectation of cost per order is found by sampling the processing of orders within a computer simulation model for a particular sequencing rule. Each sequencing rule is determined by the coefficients in a priority function which is a weighted combination of identified decision variables The search procedure thus tests different sequencing rules by varying the coefficients in the priority function and generating associated cost expectations through simulation. Rather than leading to a "single best rule" for all job shops, the appreach is a "method for finding" a sequencing rule for any specific situation.

*The authors wish to acknowledge the initial funding of this project by the University Grants Program at Arizona State University.

KEY WORD INDEX

Below is a list of key words for the papers and abstracts of this volume. It can be used to zero in on papers which are topically related to an area of the reader's interest. For example, under "Monte - Carlo Simulation" we find reference to two papers in Session 9, Financial Models, by Brewerton and Baumler and two papers in the paper fair namely those by Wallace and Bowen. Tutorials have not been referenced in this index. An interesting view of the material presented in this conference is afforded by the frequency distribution of uses of key words. Aside from a few which are universal (i.e., "Simulation") and which have been suppressed, there is very little cross referencing between papers. This can be seen from the fact that the modal reference is 1. Viewed positively we might say that this shows that the Winter Simulation Conference has not become institutionalized to the point of stifling variety.

Accounting measures - 9 Baumler
Accounting rate of return - 9 Baumler
Aerospace - PF Moore
Aggressiveness - 1 Wyman
Air traffic control - 8 Flanagan
Air pollution - PF Greene
Aircraft collision avoidance - 8 Flanagan
Airport - 10 Siesennop
Ambulance dispatching - PF Siler
Analysis of risk - 9 Brewerton
Analysis of variance - 2 Bernard
ANOVA - 1 Menke
Anti-air warfare - 8 Andrus
APL vs. FORTRAN - 11 Courtney
Arrival distribution - 2 Bernard
Artificial intelligence - 14 Heidorn
Assignment rule - 6 Maggard
Associate processor - 8 Flanagan
Assumptions - 2 Bernard
Automation - 15 Bøe
Automated warehouse - 3 Bafna
Baggage - 10 Siesennop
BASS - 3 Bafna
Behavioral adjustment - 1 Wyman
Benchmarking - 6 DeMaire
Bill of material processor - PF Shiveley
Biological simulation - PF Herman
Budget simulation - 11 Courtney
Budget - 12 Mock
Business planning - 14 Zant
Capital recovery depreciation - 9 Baumler
Cased goods - 3 Heimburger
Cash flow - PF Robinson
CDC 6000 - 14 Ast
Clinical applications - PF Delaney
Clinics Model - PF Cohn
Cognitive processes - 1 Tuggle
Combined simulation - 14 Hurst
Commodity market hedging - PF Markland
Complex systems - PF Silver
Compliance report - PF Babiel
Compound-interest amortization - 9 Baumler
Computer models - 9 Wolfe
Computer network - PF Collins
Computer simulation - 7 Schweizer, 8 Andrus
Computer systems - 15 Bøe, PF Vavra, PF Grant
Computer system design - PF Lyons
Confidence intervals - 2 Bernard
Confidence intervals in simulation - 5 Crane
Continuous-time system - PF Bowen
Control strategies - PF Lee
Control system - 8 Mitome
Conventional risk analysis - 9 Brewerton
Conveyor - 3 Heimburger
Conveyors - 3 Bussey
Corporate environment - PF Markland
Corporate models - 14 Zant
Corporate modeling - 9 Wolfe

Corporate model design - 11 Courtney
Corporate planning - 14 Zant
Crosstalk - 15 Jain
Data arrays - 14 Ast
Data banks - 14 Ast
Decision - 12 Churchill
Decision-making - 1 Tuggle, 8 Landstra, PF Eldin
Decision policies - PF Parker
Delay-line model - PF Lee
Demographic model - 7 Schweizer
Dental procedures - PF Dilworth
Design of financial models - 11 Courtney
Design of manufacturing system - 8 Mitome
Deterministic model - 8 Landstra
Digital - PF Testa
Discrete simulation - 1 Tuggle
Discrete system simulation language - 14 Ast
Distribution - PF Silver
Dynamo - PF Cohn
Ecology - 1 Menke
Education - 12 Churchill
Educational - 12 Mock
Educational planning - 7 Schweizer
Education--police training - 12 Serway
Electromechanical - PF Moore
Emulation - 8 Flanagan
Environmental engineering - PF Lee
Environmental reaction - 1 Wyman
Epidemic - PF Lee
Episode - PF Greene
Equipment layout - 8 Mitome
Equipment selection - PF Lyons
Estimation - 5 Crane
Expected profit-loss - 13 Miller
FAA specifications - PF Parker
FCFS - 6 Maggard
Financial forecasting - 9 Wolfe
Financial model design - 11 Courtney
Financial models - 12 Mock
Financial modeling - 9 Wolfe
Financial planning and controls - 12 Mock
Flow simulator - 6 DeMaire
Forecasting - 8 Landstra
French flag problem - PF Herman
Game - 12 Churchill, PF Greene
Gaming - 12 Serway
GERT - 6 Maggard
GPSS - 2 Bernard, 14 Ast, 14 Heidorn, PF Delaney,
    PF Dilworth, PF Garcia
GPSS applications - 6 Patel
Great Lakes - 10 Rea
Health care - PF Siler, PF Cohn
Health care delivery - PF Delaney, PF Dilworth
Health care planning - 7 Schweizer
High-rise warehouse - 3 Bafna
Hospital - PF Durham
Improvements in systems performance - 4 Baron
Information flow - PF Eldin
Information processing theory - 1 Tuggle
Interactive - 12 Mock
Interactive computer models - 9 Wolfe
Interactive models - 13 Blumenthal
Interactive simulation - PF Lee
Internal rate of return - 9 Baumler
Interrelationships--design, operation system,
    language - 11 Courtney
Inventory - PF Durham, PF Shiveley
Investment decisions - PF Robinson
Investment factors - 9 Brewerton
Iterative array - PF Herman
Job sequencing - PF Babiel
Job shop - 6 Franklin, 6 Maggard, PF Hershauer
Labor blocking - 6 Maggard
Labor limited - 6 Maggard
Languages for simulation - 14 Hurst
Large scale models - 14 Ast
Law enforcement--police - 12 Serway
Locks - 10 Rea
Machinery plant - 15 Bøe
Machine independent language - 14 Ast
Maintenance - PF Parker
Man-machine interactions on manufacturing line -
    6 Patel
Management game - 12 Serway, PF Lyons

Management information systems - PF Wallace, PF Eldin
Management involvement - 6 DeMaire
Manufacturing cycle-time simulation - 6 Patel
Manufacturing floor control - PF Babiel
Manufacturing system - 8 Mitome
Marine - 10 Graff
Markov chains - 5 Crane
Markov process - 15 Bøe
Maslow's need theory - 1 Wyman
Materials handling - 3 Bussey, 10 Siesennop, PF Silver
Material requirements planning - PF Shiveley
Medical facility - PF Garcia
Model design characteristics - 11 Courtney
Models, utility - 3 Bussey
Modeling - 1 Menke
Modeling philosophy - 4 Baron
Modular - 12 Churchill
Modular design - 11 Courtney
Modular simulation - PF Collins
Monte Carlo simulation - 9 Brewerton, 9 Baumler, PF Wallace, PF Bowen
Motivation - 1 Wyman
Multi-level - 12 Churchill
Multiprogramming - PF Vavra
Multiphasic screening - PF Delaney
Multiple aircraft defensive test environment - 8 Dominiak
Multiple random number streams - 2 Mize
Natural gas - 13 Blumenthal
Natural language - 14 Heidorn
Need satisfaction - 1 Wyman
Network - PF Vavra
Non-linear programming - PF Durham
Objectives - 6 Franklin
Occurence distributions - 6 DeMaire
Ogive - 13 Blumenthal
Operations research - PF Durham
Outpatient clinic - 4 Baron
Parallel processing - 8 Flanagan
Performance measurement - PF Collins
Personality simulation - 1 Wyman
Pest management - 1 Menke
Petroleum - 10 Graff
PL/1 - PF Siler
Planning - 8 Landstra, 9 Wolfe, PF Hopkins
Planning systems - 14 Zant
Poisson - 2 Bernard
Police training game - 12 Serway
Port - 10 Graff
Price level adjustments - 9 Baumler
Priority rule - 6 Maggard
Probability distributions - 9 Brewerton, PF Vasicek
Probabilistic supply-demand curves - 13 Miller
Process generators - 2 Mize
Production control - 8 Mitome
Production scheduling - PF Hopkins, PF Moore
Project selection - PF Wallace
Queueing - 10 Graff, PF Vavra
Queueing simulation - 5 Crane
Radar network target generation - 8 Dominiak
Radar - PF Bowen
RANDOM - 6 Maggard
Random number generators - 2 Mize
Random variables - PF Vasicek
Rate of return function - 9 Brewerton
Real time - 2 Bernard
Reliability - 15 Bøe
Research and development - PF Robinson
Resource allocation - 8 Landstra, PF Eldin
Resource planning - 6 Patel
Revenue forecasting - 13 Blumenthal
Risk analysis - PF Wallace, PF Vasicek, PF Robinson
Risk-return decision criterion - PF Markland
Robustness - PF Vavra
Sample size - PF Hershauer
Scheduling algorithm - 8 Mitome
Scheduling - 6 Parker, 6 Franklin, PF Hershaur
Schistosomiasis - PF Lee
Search techniques - PF Hershauer
Sensitivity analysis - 2 Bernard, PF Vavra, PF Vasicek
Sequencing - PF Hershauer

Ship - 10 Graff
Ships - 15 Bøe
Shipping - 10 Rea
Simplex algorithm - PF Jain
SIMSCRIPT - 10 Rea
SIMSCRIPT II - 1 Wyman
SIMULA 67 - 2 Vaucher
Simulation algorithms - 2 Vaucher
Simulation experiments - 1 Menke
Simulation model - 4 Baron, 6 Patel
Simulation of financial plan - 11 Courtney
Simulation programming - 14 Heidorn
Simulator - 8 Mitome
Socialization - 1 Wyman
SOT - 6 Maggard
Stability - PF Lee
Stacker cranes - 3 Bafna
Stacker crane simulator - 3 Bafna
Staffing - PF Garcia
Statistical analysis of simulations - 5 Crane
Statistical modeling - 15 Jain
Steel production - PF Jain
Stochastic networks - PF Robinson
Storage rack design - 3 Bafna
Surface-to-air missile systems - 8 Andrus
Surgical service - PF Delaney
System software testing - PF Collins
Systems analysis - 14 Ast
Systems, materials handling - 3 Bussey
Systems, multiserver - 3 Bussey
Systems simulation - 14 Ast, 14 Hurst
Systems, recirculating (feedback) - 3 Bussey
Terminal - 10 Graff
Throughput - PF Lyons
Timesharing - 9 Wolfe
Timesharing vs. batch - 11 Courtney
Trace driven - PF Grant
Training - 12 Churchill
Triage - PF Garcia
True yield - 9 Baumler
Uncertainty - 1 Tuggle
Uncertain returns - 9 Baumler
Unit time advance - 1 Wyman
Urban planning - 7 Schweizer
Utility - 15 Bøe
Utilities - 13 Blumenthal
Utility investment simulation - 13 Miller
Validation of model - 4 Baron
Variability assumption - 9 Brewerton
Variance reduction - 2 Mize
Vehicular traffic control - PF Testa
Warehouse costs - 3 Bafna
Warehouse design - 3 Bafna
Warehouse operations - 3 Bafna
Warehouse simulator - 3 Bafna